Visual Basic
للمبتدئين

أعداد المبرمج:مشتاق طالب رشيد العامري
MUSHTAQ_TALIB58@YAHOO.COM

2009

شركة الأميرال للخدمات البرمجية المتطورة/الراعي الرسمي



بسم الله الرحمن الرحيم

Visual Basic

الجزء الاول

الاساسيات

الفصل الاول

Visual Basic

. تختلف وجهات النظر بين Visual Basic هل هي
عبارة عن مكتبة من Visual Basic برنامج تصميم نوافذ ام منتج متكامل؟ تقنيا،
المكتبات توفر عدة اوامر برمجية متكاملة مع ابنتها– Visual Basic for Application
لغة برمجة Visual Basic . الا ان المتعارف عليه بين المبرمجين هو ان VBA تختصر
وبرنامج تصميم نوافذ ومكتبات تشغيلية، أي باختصار هو منتج متكامل . لذلك، عندما
في هذا الكتاب فانني اقصد جميع التعريفات السابقة، فلو Visual Basic اتحدث عن
ولو __________ ، BASIC فانني اقصد اللغة الحنونة Visual Basic ذات مزايا لغة
التشغيلية، ولو Visual Basic فانني المح الى مكتبات Visual Basic دوال واجراءات
آما في الفقرة التالية - فانني– Visual Basic تطرقت الى بيئة التطوير المتكاملة ل
اقصد برنامج تصميم النوافذ.

IDEبيئة التطوير المتكاملة

Windows برامجك ومشاريعك تستطيع انجازها باستخدام ابسط برنامج يأتي مع
ولله الحمد- لم اقابل ذلك الشخص الذي–. الا انني Notepad والذي يدعى
Visual Basic . لان Visual Basicآي يصمم برنامج ينفذ تحت Notepad يستخدم ال
يوفر لك ادوات في قمة الروعة مجتمعة تحت مسمى بيئة التطوير المتكاملة
- توفر لك آل ما تحتاجه IDE تختصر– Integrated Development Environment
خمس نجوم - آآثر م ن–لتصميم نوافذ وكتابة آاواد برامجك . بل تقدم لك خدمات
، ادارة ملفات مشروعك، تحرير القوائم، تعديل Debugging ذلك، آخدمة التنقيح
.1- 1 وإنشاء قواعد البيانات ... الخ **شكل**
3
:Visual Basic بيئة التطوير المتكاملة ل **.1- 1 شكل**
- فمن الضروري التمعن في VB5 أي لم تستخدم–اذا آنت جديدا على هذه البيئة
آلمات هذا الفصل بالذ ات والا فانك لن تدل الطرق المؤدية الى ساحة العمل .
على ما اعتقد –-الطرق ستكون نوافذ بيئة التطوير ، والعمل هو السبب الذي جعلك
تقرأ هذا الكتاب.
، تحتوي بيئة التطوير على الكثير من النوافذ المحضونة**1- 1** آما تلاحظ في **شكل**
ومجموعات من الازرار تدعى Menus والعديد من القوائم Child Windows بها
، واليك يا سيدي تفاصيلها: Toolbars اشرطة الادوات

نوافذ بيئة التطوير المتكاملة

الاصدار الخامس فانتقل الى فقرة "نافذة Visual Basic اذا آنت من جيل مبرمجي
" لانه لا يوجد أي شئ جديد اوضحه لك هنا. اما ان آنت Data View عرض البيانات
من الاجي ال الاقدم، فاول شئ قد يشد انتباهك هو أن النافذة الرئيسية لبيئة
التطوير اصبحت من النوع متعدد المستندات– Multiple Document Interface
، وستلاحظ اشتمالها على نوافذ جديدة ، بالاضافة إلى تطوير بعض MDI تختصر

النوافذ السابقة نافذة مستكشف المشروع او الخصائص . واول ناف ذة سنبدأ بها
هي من النوافذ الجديدة:
4
New Projectنافذة مشروع جديد
لاول مرة، فان لهذه النافذة احتمال Visual Basic عندما تقوم بتشغيل منصة العمل
مؤآد للظهور. فعن طريقها تستطيع الاختيار بين أنواع عدة من المشاريع آ
ثم Standard EXE .... الخ. حدد النوع Standard EXE ،ActiveX EXE ،ActiveX DLL
]. فالقاعدة السائدة لجميع ENTERاو اضغط على المفتاح [Open انقر على الزر
ولن اشذ عن Standard EXE التعريفية تبدأ الشرح دائما بالنوع Visual Basic أكتب
هذه القاعدة، فانت لا تريد معرفة طريقة طبخ الكبسة قبل ان تتعلم قلي البيض!
Visual اذا أتت هذه النافذة تسبب لك ازعاج -ولو بسيط- في آل مرة تشغل فيها
Basic ، تستطيع الغاء فكرة ظهورها عن طريق تحديد الخيار Don't show this
الموجود في اسفلها حتى لا تعكر مزاجك، وتكون رفيقا لي dialog in the future
حتى نهاية الكتاب.
ملاحظة: إذا لم يرق لك هذا التغيير، واردت العودة إلى الوضع السابق، فاختر
، ومن مربع الحوار الذي يظهر Tools من قائمة Options ...الأمر
Prompt for ، وانقر على الخيار Environment امامك اختر تبويب
.project
Form Designerنافذة مصمم النماذج
منذ Visual Basic ، وهي اشهر نو افذ Visual Basic هذه النافذة تعتبر سر نجاح
. عن طريق هذه النافذة تقوم بعملية Form1 الاصدار الاول مع العنوان الابتدائي لها
تصميم واجهة برنامجك اما بتعديل خصائصها او وضع باقة ادوات عليها باستخدام
Visual مبينة لك قوة ونجاح فكرة لغات البرمجة المرئية Mouse الفأرة
. فقد انتهى عصر تصميم الشاشات بالآواد او تعليماتProgramming Languages
Tapesواشرطة التسجيل . 5.25 آما آنا نفعل في زمن اقراص Macro الماآرو
طريقة وضع الادوات على نافذة النموذج اشبه ما تكون بعملية رسم مربعات آما
. المزيد ايضا، عمليات التحرير آالنسخ واللصق و القص Paint في برنامج الرسام
مدعومة على جميع الادوات التي تضعها.
ملاحظة: نافذة النموذج هي اسم مختصر من نافذة مصمم النماذج والمقصد
واحد. لذلك، ستلاحظ انني استخدم المصطلح نافذة النموذج في
آافة ارجاء الكتاب .
5
Toolboxنافذة صندوق الادوات
بعد ان تظهر لنا نافذة النموذج السابقة، فان شهوة وضع الادوات عليها تصل الى
التي **2- 1 شكل** Toolbox قمتها. والادوات موجودة في نافذة صندوق الادوات
Visual Basicأداة قياسية مدمجة في جميع برامجك المصممة تحت ، 20 تعرض لك
-مازال الوقت ActiveX Controls وقد تحتوي على مجموعة ادوات اضافية تدعى
Buttonsزر ، 21 أداة الا انه يوجد 20مبكرا جدا للحديث ع نها. مع ان الادوات عددها
هذا الزر الاضافي موجود في الرآن العلوي الايسر من الادوات على شكل مؤشر
وظيفته الاساسية الغاء عملية طلب رسم أداة . لا تشغل نفسك به آثيرا، Pointer
فهو يضغط نفسه تلقائيا بمجرد انتهائك من عملية رسم او وضع الأداة على نافذة
النموذج.
:Toolbox صندوق الادوات .**2- 1 شكل**
اذا أن عدد الادوات الموجودة في صندوق الادوات آبيرا جدا، فيفضل ان تقوم
بعملية تقسيم الادوات الى مجموعات تختفي وتظهر متى ما شئت عن طريق
من Add Tab ...النقر بزر الفأرة الايمن على نافذة صندوق الادوات واختيار الامر
القائمة المنسدلة ومن ثم كتابة اسم المجموعة . طريقة تنظيم محتويات آل
وهو نفس الاسلوب الذي Drag & Drop مجموعة تتبع اسلوب السحب والالقاء

تتبعه لنسخ او نقل ملفات جهازك . اخيرا، اذا اردت حذف المجموعة، قم بالنقر على
من القائمة المنسدلة، Delete Tab اسم المجموعة بزر الفأرة الايمن واختيار الامر
لن تتمكن من حذفها. General مع العلم ان المجموعة الرئيسة والتي تسمى
يبدو انني نسيت نقطة مهمة وهي آيف تعرض نافذة صندوق الادوات في حالة
Viewمن القائمة . Toolbox اغلاقها، يتم ذلك عن طريق اختيار الامر

Properties Windowsنافذة الخصائص

بمجرد انتهائك من وضع الأداة على نافذة النموذج، فان عينيك ست بحث عن موقع
نافذة الخصائص والتي من خلالها ستتمكن من تعديل خصائص الأداة او حتى نافذة

، من هذه الخصائص الحجم، الموقع، اللون، العنوان .... الخ. اذا أنت Form النموذج
او اضغط على View من قائمة Properties Window هذه النافذة مخفية اختر الامر
F4المفتاح [ ].
ComboBox بأداة ال Visual Basic في اعلى النافذة يوجد قائمة تسمى في عالم
تمكنك من تحديد الكائن او الأداة التي تود عرض خصائصه ا. بامكانك تحديد الأداة
مباشرة بالنقر عليها -وهي على نافذة النموذج- وستلاحظ ان محتويات نافذة
الخصائص قد تغيرت . المزيد ايضا، يمكنك اختيار طريقة ترتيب جدول الخصائص اما
. وبالنسبة للجدول، فان العمود Categorized او مصنف Alphabetic ترتيب ابجدي
الايسر يعرض لك الخصائص المتوفرة للأداة اما الايمن فيعرض قيمة آل خاصية من
هذه الخصائص . بعض الخصائص تستطيع تعديلها مباشرة بكتابة قيمة عددية او
او لون من Visible ، وبعضها عليك اختيار قيمة من عدة قيم آ Caption حرفية آ
، وهناك نوع يظهر لك مربع صغير في اقصى BackColor مجموعة لوح الالوان آ
له Dialog Box يمين العمود مكتوب عليه ثلاث نقاط "..." يقصد به ص ندوق حوار
Fontخيارات اضافية آالخاصية .

Project Explorerنافذة مستكشف المشروع

تزداد اهمية هذه النافذة بازدياد عدد الملفات التابعة لمشروعك، فهي الوسيلة
الوحيدة التي تمكنك من عرض محتويات مشروعك مرتبة على شكل شجري برموز
مختلفة تجد شرحا مفصلا له ا في ملف التعليمات . تستطيع الوصول الى الصفحة
]. ان لم F1التي اقصدها عن طريق تحديد النافذة ومن ثم الضغط على المفتاح [
تكن نافذة مستكشف المشروع ظاهرة امامك، تستطيع عرضها باختيار الامر
Ctrl +Rاو الضغط على المفاتيح [ ]. View من قائمة Project Explorer

Code Windowنافذة محرر الاآواد

لغة برمجة، فبكل تأكيد عليك كتابة آاواد وتعليمات اللغة . عن Visual Basic بما ان
طريق نافذة محرر الاآواد يمكنك عمل ذلك، فه ي توفر لك محرر برمجي ذآي جدا
ومنسق آلمات يفتح نفس المبرمج لكتابة الاآواد . من المناسب ان انوه هنا بان
بعض المبرمجين العرب الذين يستخدمون نظم تدعم مجموعة محارف الشيفرة
يواجهون احيانا مشاآل في كتابة Windows 2000, XP آ Unicode الموحدة
الحروف العربية، والسبب خارج نظاق الفقرة لكن حلها يتم عن طريق تغيير نوع
. يمكنك عم ل Arabic(Courier Newالي خط يدعم اللغة العربية آ (Font الخط
الموجودة في صندوق الحوار Editor Format ذلك بالتوجه الى خانة التبويب
Toolsمن قائمة . Options والذي تصل اليه عن طريق اختيار الامر Options

بامكانك مشاهدة نافذة محرر الاآواد اما بالنقر المزدوج بالفأرة على الأداة او نافذة
]. اخيرا، اذا آنت لا تحب مشاهدة افلام F7النموذج، او بالضغط على المفتاح [
الرعب البرمجي المتمثلة في الاآواد الطويلة، تمكنك هذه النافذة من عرض اجراء
الموجود في Procedure View معين واخفاء سائر الاآواد عن طريق النقر على الزر
الرآن السفلي الايسر من النافذة، لكن عملية التنقل بين الاجراءات ستكون مملة
الموجودتان في اعلى النافذة، ComboBoxes بعض الشئ عن طريق القائمتين
القائمة اليسرى تعرض جميع الادوات الموجودة في نافذة النموذج الحالية، بالاضافة

) وهي تشير إلى قسم الاعلانات في Generalإلى النموذج نفسة، وآذلك العبارة (
النموذج وآذلك الاجراءات والدوال التي تقو م بانشاءها، والقائمة اليمنى تعرض
جميع الاجراءات والاحداث المرتبطة بما يتم اختيارة في القائمة الاولى.

Form Layoutنافذة مخطط النموذج

يؤدي الى ظهور هذه View الموجود في قائمة Form Layout Window اختيار الامر
النافذة والتي تعطيك رؤية مبسطة عن موقع وحجم نافذة النموذج التي تصممها
وقت التنفيذ من الشاشة . الا ان الفائدة الكبرى التي تجنيها من هذه النافذة تكون
المختلفة للشاشة . Resolutions مقارنة حجم نافذة النموذج مع الكثافات النقطية
Resolution لعرض هذه الكثافات، انقر بزر الفأرة الايمن على النافذة واختر الامر
من القائمة المنسدلة. Guide

Immediate Windowنافذة موجة الاوامر

بسيطة للاختبار Visual Basic تمكنك هذه النافذة من كتابة اوامر وتعليمات لغة
والتجربة، قد تحتاجها مثلا لاختبار امر معين او قيمة متغير معين قبل وضع الكود له .
] يؤدي الى الى تنفيذ الامر . تستطيع عرض ENTERبعد ان تكتب الامر، المفتاح [
Viewمن قائمة . Immediate Window هذه النافذة باختيار الامر

Object Browserنافذة مستعرض الكائنات

اذا آنت مبتدءاً فإن هذه النافذة لن تثير اهتمامك ، ولكن بعد ان تصل الى مرحلة
سيكون لهذه النافذة تقدير آ بير . تعرض لك Visual Basic متقدمة من البرمجة ب
هذه النافذة جميع الفئات الموجودة في المكتبات المضمنة في برنامجك مع آآفة
طرقها، خصائصها واحداثها لتعطيك فكرة عامة عن محتويات هذه المكتبات . ميزة
اخرى استخدمها كثيراً هي انها تسهل عليك عملية ايجاد صفحة التعليمات الخاصة
بالامر الذي تريده عن طريق ا لنقر بزر الفأرة الايمن على العنصر المطلوب واختيار
من القائمة المنسدلة . تستطيع عرض نافذة مستعرض الكائنات باختيار Help الامر
F2او الضغط على المفتاح [.] View من قائمة Object Browser الامر

Local Windowنافذة المتغيرات المحلية

، لانها في وقت Run Time الغرض الرئيسي من هذه النافذة يظهر في وقت التنفيذ
ليس لها أي وظيفة ايجابية فلا تعرضها حتى لا تتزاحم Design Time التصميم
Local النوافذ امامك. تعرض لك هذه النافذة قيم جميع المتغيرات المحلية
الحالي والذي يتم تنفيذه اذا قمت Procedure الموجود في الاجراء Variables
محلية في Objects للبرنامج . في حالة وجود أسنادات Pause بعملية الا يقاف المؤقت
الاجراء، فان اسم الكائن سيظهر في الجدول ملاصق لعلامة الزائد "+" حتى تقوم
بالنقر عليه ويعرض لك جميع الخصائص وقيمها التابعة لهذا الكائن . لعرض هذه
Viewمن القائمة . Local Window النافذة اختر الامر
–هو المرحلة التي يتم تنفيذ برنامجك فيها Run Time **ملاحظة:** وقت التنفيذ
Design Time]، اما وقت التصميم F5بعد الضغط على المفتاح [
فهي المرحلة التي تصمم برنامجك به ا. يوجد وقت خاص يعرف
وهو باختصار عملية Break او القطع Pause Time بالايقاف المؤقت
الوقف المؤقت لتنفيذ برنامجك ولكن ليس انهائه.

Watches Windowنافذة المراقبة

تمكنك هذه النافذة من مراقبة المتغيرات او العبارات التي تضيفها به ا. عملية
المراقبة تكون متواصلة ودورية التحديث آوزارة الاعلام . بامكانك تجميد عملية تنفيذ
، او True الايقاف المؤقت - في حال آون قيمة المتغير اصبحت صحيحة–البرنامج
عند أي تغيير يطرأ على قيمة المتغير . الهدف من نافذة المراقبة ليس آالهداف وزراة
ومعرفة قيم المتغيرات التي Debugging الاعلام، وانما تسهيل عملية التنقيح
البرامج. لعرض هذه النافذة اختر الامر Bugs % من اسباب شوائب 90 تشكل نسبة
، ولاضافة متغير او قيمة لجعلها خاضعة تحت View من قائمة Watch Window
- او انقر بزر View وليس– Debug من قائمة Add Watch ...المراقبة، اختر الامر

الفأرة الايمن على النافذة ومن ثم اختيار نفس الامر من القائمة المنسدلة . الميزة الموجودة في هذه النافذة تغطى على القصور الموجود في النافذة السابقة، لان Globalهذه النافذة تسمح باضافة المتغيرات العامة .

Call Stackنافذة استدعاء الاجراءات

تتشابه هذه النافذة مع النافذتين السابقتين في أنها نوافذ خاصة ب التنقيح. تعرض هذه النافذة طابور للاجراءات التي سيتم استدعائها، تفيدك هذه النافذة بقوة في . اخيرا، بامكانك عرض هذه Recursion حالة استخدامك للاستدعاءات التراجعية

9

النافذة وقت الايقاف المؤقت في حال وجود اجراءات منتظرة الاستدعاء عن طريق Ctrl+Lاو الضغط على المفاتيح [.View] الموجود في قائمة Call Stack الامر

Data View نافذة عرض البيانات

من قائمة Data View Window تستطيع عرض هذه النافذة عن طريق اختيار الامر . تعتبر Standard Toolbar او النقر على رمزها في شريط الادوات القياسي View ، فلم IDE نافذة عرض البيانات احدث نافذة دخلت صندوق بيئة التطوير المتكاملة وهو السادس . هذه النافذة Visual Basic تظهر الا في الاصدار الاخير من اصدارات دون مبالغة –البسيطة في مظهرها تعتبر اقوى نافذة من النوافذ التي ذآرته !! لانها تعتبر برنامج آامل لمراقبة، إنشاء، تعديل، حذف وعرض جداول وحقول قواعد .... الخ. MS-SQL Server ،MS-Access ،ORACLEالبيانات باختلاف أنواعها مثل: لذلك، فانه من البديهي ان لا نضيع وقتنا ونفصلها الان، فالجزء الثاني من اجزاء الكتاب "برمجة قواعد البيانات" هو اهل للتفصيل -والتطريز اذا آنت تريد!

قوائم بيئة التطوير المتكاملة

اذا تحدثت عن آل امر من اوامر قوائم بيئة التطوير المتكاملة في هذه الفقرة، لي ولك - ان يتم شرح وظيفة–. فيستحسن MSDN فسيكون الكتاب اشبه بم راجع الامر في حين استخدامه، الا انني سأعرفك على القوائم بطريقة مبسطة:

Fileالقائمة :

تحتوي هذه القائمة على اوامر اساسية خاصة للمشاريع بشكل عام، أنشاء مشروع جديد، حفظ محتويات المشروع، طباعة محتويات المشروع وترجمة هي امكانية VB6 . الميزة التي اضيفت ل Binary المشروع وتحويله الى ملف ثنائي فتح آثر من مشروع في نسخة واحدة من البيئة، وهي ميزة تعرف بالمشاريع Multiple Projectsالمتعددة .

Editالقائمة :

تحوي هذه القائمة على اوامر التحرير القياسية آالقص، النسخ واللصق . بالاضافة الى اوامر خاصة بقواعد البيانات في حالة وجود قاعدة بيانات في نافذة عرض . معظم الاوامر الواردة في اسفل هذه القائمة تستخدمها مع Data View البيانات Code Windowنافذة محرر الآواد .

10

Viewالقائمة :

ذآرت معظم محتوياتها في فقرة "نوافذ بيئة التطوير المتكاملة "، وبالا عتماد على نباهتك ومدى استعيباك يمكنك معرفة الغرض هذه القائمة.

Projectالقائمة :

معظم اوامرها خاصة بمحتويات المشاريع، فهي تمكنك من اضافة عنصر او عناصر Classes ، فئات Module ، ملفات البرمجة Forms من عناصر المشروع آنوافذ النماذج اضافية عن ActiveX Controls .... الخ. المزيد ايضا، يمكنك اضافة ادوات تحكم خارجية عن طريق الامر ActiveX DLL او تضمين مكتبات Components طريق الامر .References

Formatالقائمة :

الاوامر الموجودة في هذه القائمة خاصة بتنسيق الادوات التي تضعها على نافذة النموذج من ناحية موقعها على النافذة، فتوجد اوامر مرنة توفر عليك جهد محاذاة

الادوات او توسيطها على النافذة، بالاضافة الى تغيير ترتيب ظهور الادوات أي وضع
تستخدمه Lock Controls أداة فوق الكل او أداة خلف الكل . الامر الاخير
اذا آنت___________
راضيا عن تصميم الادوات وتود منع نفسك من تغيير احجامها او مواقعا عن طريق
الخطأ، هذا القفل تستطيع فتحة بكل بساطة باختيار نفس الامر مرة اخرى.

Debugالقائمة :

معظم اوامر التنقيح وضعت اسفل هذه القائمة . من هذه الاوامر اختيار طريقة تنفيذ
، امر سابق Step Over ، اجراء آامل Step Into البرنامج، آتنفيذ سطر واحد منه
Run او التنفيذ حتى الوصول الى السطر الذي يوجد عليه مؤشر الكتابة Step Out
فهي علامات تظهر مبدئيا باللون BreakPoints . وبالنسبة لنقاط القطع to Cursor
الاحمر على سطر معين بحيث تتم عملية الايقاف المؤقت للبرنامج عند الوصول
الى هذه العلامات.

Runالقائمة :

عن طريق هذه النافذة البسيطة تست طيع تنفيذ البرنامج وتتمكنك من اختيار الاوامر
. بالنسبة للامر End او انهاء عملية تنفيذ البرنامج Break الاخرى آالايقاف المؤقت
ولن تحتاجه الا في حالات Start هو مشابه لامر التنفيذ Start with Full Compile
نادرة ستجدها لاحقا في هذا الكتاب.

11

Queryالقائمة :

وهي متوفرة لنسخة المحترفين VB5هذه القائمة جديدة على مبرمجي
VB6للاصدار السادس . Enterprise Edition والمؤسسات Professional Edition
باستخدام الأداة SQL اوامر هذه القائمة غير ممكنة حتى تنشئ جملة استعلام
.Microsoft Query Builder

Diagramالقائمة :

ومتوفرة لنفس النسخ المذآورة في VB5 ايضا هذه قائمة جديد ة على مبرمجي
الفقرة السابقة . اوامر هذه القائمة غير ممكنة الا في حالة تعاملك مع قاعدة بيانات
ORACLEاو . SQL Server

Toolsالقائمة :

ومسهل كتابة Menu Editor تحتوي على اوامر مختلفة التصانيف آمحرر القوائم
IDE وغيرها... اذا ادرت تخصيص بيئة التطوير المتكاملة Add Procedure الاجراءات
الذي يوفر لك Options يمكنك من الوصول الى صندوق الحوار Options فالامر
IDEعشرات الخيارات والخاصة بتغيير اعدادات بيئة التطوير .

Add-Insالقائمة :

Addالاوامر الموجودة في هذه القائمة عبارة عن برامج مستقلة تسمى الاضافات -
هدفها توفير خدمات اضافية لبيئة التطوير تزيد من مرونته ا. تطوير هذا النوع من Ins
البرامج خارج نطاق الكتاب.

Windowالقائمة :

Visual اذا آنت بحاجة الى شرح محتويات هذه القائمة، فيبدو انك جديد ليس على
، فافضل شرح اسطره لك Windows فحسب وانما على جميع تطبيقات بيئة Basic
الايقاف المؤقت - في الحال، وتحاول التعرف على بيئة–هو بان توقف قراءة الكتاب
ومن ثم تعود لمتابعة القراءة. Windows

Helpالقائمة :

، فلن يتمكنوا من الوصول Visual Basic بالنسبة لمستخدمي الاصدار السادس من
Microsoft Developer الى التعليمات الفورية الا في حالة انزال نسخة من مكتبة
MSDNوالمألوفة بالاختصار . Network

12

اشرطة الادوات

جميع الازرار الموجودة في اشرطة الادوات منسوخة من القوائم السابقة، فلايوجد

داعي لاعادة الشرح. اما الغرض من نسخها فهو تسريع عملية اختيار الامر.
MS-Office تستطيع التحكم في اشرطة الادو ات وتحريرها آما تفعل مع تطبيقات
من القائمة Customize بالضغط بزر الفارة الايمن على شريط الادوات واختيار الامر
قد ادرجته ضمن فقرة ToolBox المنسدلة. احب ان انوه هنا بان صندوق الادوات
نوافذ بيئة التطوير، فلا تعتقد انه من فئة اشرطة الادوات رغم الشبه الكبير بينهم.

كتابة برنامجك الاول

لاشك ان الممارسة والتطبيق احد اهم عوامل تعلم لغات البرمجة، والفصل الاول
من هذا الكتاب سيبدأ معك الممارسة ليس فقط لكتابة برنامجك الاول، وانما
لتوضيح المراحل والخطوات السليمة لعملية بناء البرامج سواء أتت شخصية او
تجارية.

الخطوة الاولى: فكرة البرنامج

ولا باي لغة اخرى ولا حتى بنظام Visual Basic فكرة البرنامج ليست لها علاقة ب
التشغيل. فمن البديهي انك قبل ان تكتب برنامج عليك معرفة ما الذي تريده من
البرنامج. قد تأتيك فكرة برنامج بينما تقلب صفحات الكتاب او متأملا النجوم في
وضح النهار، وقد تكون الفكرة الزامية عليك آحل مشكلة تصادفك في جهازك او
مشروع تخرج جامعي.
بعد ان تخطر الفكرة في بالك وتظهر لمبة مضيئة فوق رأسك، اسحب اللمبة ورآبها
في الابجورة المجاورة لجهازك، حتى ترآز وتقوم بعملية التخطيط لتطبيق الفكرة
، حدد المتطلبات والمشاآل التي Visual Basic على الارجح - ب–آبرنامج يصمم
Software ستصادفك. مثل هذه الامور تندرج تحت علم هندسة البرامج
والتحدث عنها خارج نطاق الكتاب . لذلك، ساختم هذه الفقرة بالفكرة Engineering
التي سنقوم بتنفيذها وهي عبارة عن برنامج يجري عملية الضرب بين عددين.

الخطوة الثانية: إنشاء المشروع

بعد تحديد فكرة البرنامج وتوضيح المتطلبات سنبدأ بإنشاء ملفات المشروع . قم
. هذا النوع Standard EXE ثم حدد النوع File من قائمة New Project باختيار الامر
13
، وعند Windows من المشاريع هي مشاريع بناء برامج قياسية تعمل تحت بيئة
EXEالترجمة تكون ملفاتها من النوع .
بعد قيامك بعملية إنشاء المشروع الجديد، قم بكتابة اسم مناسب للمشروع في
Project Properties... الموجودة في صندوق الحوار Project Name خانة النص
. اآتب اسم View من قائمة Project1 Properties والذي تصل اليه باختيار الامر
MyFirstProgramابتدائي للمشروع وليكن .
ملاحظة: يكون عنوان الامر السابق مختلف من اسم مشروع الى آخر . فبعد
MyFirstProgram تعديلك لاسم المشروع، سيكون عنوان الامر
.Properties...
لحفظ ملفات المشروع، تذآر ان المشروع File من قائمة Save Project اختر الامر
خاص بملفات المشروع Folder يحتوي على عدة ملفات لذلك يفضل انشاء مجلد
، اما الملفات الاخرى فهي VBP قبل حفظه . ملف المشروع الرئيس يكون امتداده
.*، ملفات BAS .*، ملفات البرمجة FRM عناصر مكونة للمشروع آنوافذ النماذج
.* .... الخ. CLS الفئات

الخطوة الثالثة: تصميم الواجهة

يندرج تحت صنف لغات البرمجة المرئية، فعملية تصميم Visual Basic بما ان
الواجهة تتم في الغالب باستخدام الفأرة . نبدأ في العادة بوضع الادوات على نافذة
النموذج ومن ثم تعديل خصائصه ا. عملية وضع الادوات على نافذة النموذج تتم
بشكل مشابه لرسم المربعات في برامج الرسم، واذا آنت تواجه صعوبة في عمل
Paintذلك، فدرب نفسك بضع دقائق على برنامج الرسام .

و زر اوامر Label وضع أداة عنوان ToolBox انتقل الى صندوق الادوات
لرسم الخط، ومن ثم رتب Line والأداة TextBoxes واداتي CommandButton
:3- 1 الادوات بشكل مشابه **للشكل**
: واجهة النافذة الرئيسة لبرنامجك الاول.
14
بعد ترتيب الادوات ومحاذاتها سنبدأ بعملية تعيين خصائصه ا. حدد الأداة بالنقر عليها
وانتقل الى نافذة الخصائص وقم بتعديل قيم خصائص الادوات آما في الجدول
التالي:
الاداة الخاصية القيمة
نافذة النموذج
زر الاوامر
أداة العنوان
أداتي النص
Name
Caption
Name
Caption
Name
Caption
Name
Caption
Name
Caption
frmMain
"البرنامج الاول"
cmdMultiply
"اضرب"
lblProduct
"0"
txtFirst
"0"
txtSecond
"0"
كتابة حروف بادئة قبل اسم الأداة Visual Basic **ملاحظة:** من تقاليد مبرمجي
frm بحيث يستدل المبرمج على نوع الاداة عن طريق الأسم آ
لخانة النص .... الخ، وستلاحظ txt لزر الاوامر، cmd لنافذة النموذج،
انني أكتبعها بشكل جليل في الكتاب لتمسكي بعادات وتقاليد
البرمجية. Visual Basic احزاب
وبذلك نكون قد انتهينا من الخطوة الثالثة: تصميم الواجهة.

الخطوة الرابعة: كتابة التعليمات

بعد تصميم الواجهة والاقتناع بمظهرها الفاتن، تبدأ خطوة كتابة التعليمات او الآواد
أي خطوة البرمجة الفعلية . قم بالنقر المزدوج على زر الاوامر، ستلاحظ ان نافذة
قد انيرت وظهر بها هذا الكود: Code Window محرر الآواد

```
Private Sub cmdMultiply_Click()
End Sub
```

15
أي آود تكتبه بين ا لسطرين السابقين سيتم تنفيذه اذا ما قام المستخدم بالنقر
. وهذه فلسفة cmdMultiply التابع للأداة Click على زر الاوامر، فهو مكان الحدث

، فالآواد لا يتم تنفيذها Event Driven Programmingالبرمجة المسيرة بالاحداث
من اول سطر الى اخر سطر آما في السابق . آحل السطر الس ابق بكتابة الكود
التالي الخاص باجراء عملية الضرب:

```
Private Sub cmdMultiply_Click()
lblProduct.Caption =CDbl)txtFirst.Text * (CDbl)txtSecond.Text(
End Sub
```

ان لم تفهم الغرض من الكود السابق، فالفصول القادمة آتية اليك، وستجد فيها
بشكل مؤقت - ان الكود السابق يقوم بعملية–الجواب الشافي، ولكن اعرف الان
ضرب القيم التي سيكتبها المستخدم في خانات النص وسيضع ناتج الضرب في
أداة العنوان.

الخطوة الخامسة: التجربة والتعديل

] لتنفيذ البرنامج. ستظهر لك نافذة النموذج التي F5قم فورا بالضغط على المفتاح [
في خانات النص ومن ثم 5 و 2 صممتها وآافة الا دوات محضونة بها، أآتب الاعداد
لامعا في أداة العنوان. 10 انقر على الزر "اضرب" آي تحصل على الناتج
اذا آنت مسرورا جدا من نجاح عملية تنفيذ البرنامج، فتذآر ان عنوان هذه الخطوة
"التجربة والتعديل " وليس "التنفيذ "، لانك ستضطر لاحقاً لاعاد ة تنقيح برنامجك
وليس–بعدما تكتشف هذه المشكلة البسيطة اذا قام المستخدم بكتابة حروف
اعداد- في خانات النص ثم قام بالنقر على الزر "اضرب "، ستلاحظ ظهور رسالة
خطأ، وسيتوقف البرنامج عن العمل . يعرف هذا النوع من الأخطاء باخطاء وقت
وستختفي به جة الفرحة بآتمال برنامجك الاول، لان Run Time Errors التنفيذ
لا يجري عملية الضرب على الحروف، فلا تتوقع ان: Visual Basic
= ترآي ترآي ترآي ترآي 5ترآي *
لذلك، عليك باعادة عملية تعديل الكود ليتحقق من القيم المدخلة من قبل
المستخدم قبل اجراء عملية الضرب عليها، ويصبح الشكل النهائي للكود المعدل
هو الكود الموجود في الصفحة التالية:

16

```
Private Sub cmdMultiply_Click()
' التحقق من القيم المدخلة
' قبل اجراء عملية الضرب عليها
If IsNumeric)txtFirst.Text = (True And IsNumeric)txtSecond.Text = (True Then
lblProduct.Caption =CDbl)txtFirst.Text * (CDbl)txtSecond.Text(
Else
MsgBox"القيم المدخلة غير صحيحة!"
End If
End Sub
```

الهدف من هذه الخطوة ليس شرح الآواد او طريقة تفادي الاخطاء، وانما توضيح
قضية التجربة والتعديل وبيان اهميتها قبل الانتهاء من تصميم البرنامج.

الخطوة السادسة: الترجمة

اتمنى ان تكون فرحة نجاح برنامجك قد عادت من جديد، الخطوة الاخيرة التي
، والمقصود منها عملية تحويل برنامجك الى Compiling سنقوم بها تعرف بالترجمة
Make ، يتم ذلك باختيار الامر EXE بالامتداد Executable ملف تنفيذي
Fileمن قائمة . MyFirstProgram.EXE
تكون عملية Compiling **ملاحظة:** في معظم لغات البرمجة الاخرى، الترجمة
Visual . اما مع EXE التي تنتج ملفات Linking قبل عملية الربط
Linkingهي نفس الربط . Compiling ، فالترجمة Basic
الميزة في عملية الترجمة هي امكانية تنفيذ برنامجك على جميع الاجهزة التي
باختلاف اصداراته والتي لا تحتوي على نسخة من Windows تحمل نظام التشغيل
، لعل Visual Basic التي تأتي مع DLL شريطة وجود بعض مكتبات Visual Basic

. لذلك، عليك ارفاق هذه المكتبة مع ملف برنامجك MSVBVM60.DLL ابرزها مكتبة
Visual Basicالى الاجهزة الاخرى والتي لا تحتوي على . EXE الرئيسي
17

الفصل الثاني

النماذج والادوات

، والنموذج Window مرادف للنافذة Visual Basic هو مصطلح خاص ب Form النموذج
هو بالفعل نافذة تقوم بتصميمها وتظهر في وقت التنفيذ آسائر نوافذ تطبيقات
فهي نوافذ ايضا، ولكن م ن نوع خاص توضع دائما Controls . اما الادوات Windows
داخل النماذج وتحتضن فيه . من المهم ان الفت النظر الى وجود نوعين من الادوات
Built-in ، النوع الاول هي الادوات الداخلية Visual Basic التي يمكن استخدامها مع
- وهي العشرين أداة الموجودة مبدئيا Intrinsic Controls -او الجوهرية Controls
Visual Basic . جميع برامجك المصممة تحت**2- 1**في نافذة صندوق الادوات
ملزمة بهذه الادوات حتى لو لم تستخدمها، فهي مضمنة في مكتبة
التي يشترط وجودها لتنفيذ برامجك . اما النوع الثاني من الادوات MSVBVM60.DLL
OCXوهي ادوات خارجية يكون امتداد ملف اتها . ActiveX Controls فتعرف بالاسم
لاستخدام هذا النوع من الادوات، عليك اضافتها الى مشروعك عن طريق اختيار
Projectمن قائمة . Components الامر
. وبما Objects النماذج والادوات تتشارك في صفة مشترآة واحدة وهي انها أسنادات
- تميز الكائنات Members انها أسنادات، فهي تتكون من ثلاث عناصر -تسمى الاعضاء
. آل Events والاحداث Methods ، الطرق Properties عن غيرها هي : الخصائص
يحتوي على اعضاء خاصة به، فالنموذج له اعضاء Visual Basic أسناد من أسنادات
مستقلة وأداة النص لها اعضاء مستقلة، الا ان النماذج والادوات تحتويان على
عشرات الاعضاء المشترآة بينه ا. لذلك، وجدت من الافضل ان ابدأ بعرض الخصائص،
الطرق والاحداث المشترآة بين النماذج والادوات، ومن ثم ذآر اعضاء آل أسناد على
حده.

الخصائص

، او في سلوك Caption الخاصية هي قيمة تؤثر اما في الشكل الخارجي للأداة آ
. نافذة الخصائص هي المكان الذي يمكنك من تغيير قيمة Enabled عمل الأداة آ
18
الخاصية وقت التصميم، اما وقت التنفيذ فعليك كتابة الكود والذي تكون صيغته
مبتدئة باسم الأداة فنقطة ثم اسم الخاصية:

```
"ترآي العامري" = Text1.Text
PictureBox1.BackColor =0
Label1.Caption =Text1.Text
```

اذا آنت ترغب تعديل مجموعة خصائص لكائن With او استخدام الكلمة المحجوزة
او اداة معينة:

```
With Text1
. = "ترآي العامري"Text
.Font.Bold =True
.BackColor =vbBlack
.ForeColor =vbWhite
End With
```

، تغنيك Default Property تتميز بعض الأدوات بوجود خاصية افتراضية لها تعرف ب
هذه الخاصية عن آ تابة اسم الخاصية بعد الأداة اذا اردت تغيير قيمتها برمجي اً.
Text والخاصية Caption معظم الادوات الداخلية تكون خاصيتها الافتراضية هي
لأداة النص:

```
"الخاصية الافتراضية" = Label1.Caption
```

Label1 = "الخاصية الافتراضية"
Text1 ="Text1.Text"
بالنسبة لنافذة النموذج، يمكنك الوصول الى خصائصها دون الالتزام بالصيغة
او حتى تجاهلها: Me السابقة فتستطيع استخدام الكلمة المحجوزة
`جميع الآواد التالية متشابهه
Form1.Caption = "النافذة الرئيسة"
Me.Caption = "النافذة الرئيسة"
Caption = "النافذة الرئيسة"
من الضروري ان انبه هنا إلى انه ليست آل الخصائص قابلة للتعديل وقت التصميم
التابعة لنافذة النموذج لايمكنك تعديلها BorderStyle والتنفيذ، فبعض الخصائص آ
19
-التابعة لنفس الكائن - لايمكنك CurrentX وقت التنفيذ، وعلى العكس الخاصية
تعديلها الا وقت التنفيذ . المزيد ايضا، هنالك بعض الخصائص غير قابلة للتعديل
، فهي خصائص تقرأها hWnd لا في وقت التنفيذ ولا التصميم - آالخاصية–مطلقا
Read Only Properties فقط وتستفيد من قيمها وهي تعرف بخصائص القراءة فقط
في وقت التنفيذ او التصميم او آلاهم ا. وعلى العكس، يوجد نوع من الخصائص
ولا تستطيع قراءتها في وقت التصميم Write Only Properties تعرف بالكتابة فقط
او التنفيذ او آلاهما.
والآن دعنا نتعرف آآثر على الخصائص المشترآة للادوات، والبداية ستكون مع
Nameخاصية الاسم .

Nameخاصية الاسم

تعديل هذه الخاصية ممكن في وقت التصميم فقط، وهي الخاصية التي تمثل
الاسم البرمجي للأداة . فالاسم الذي تكتبه في هذه الخاصية هو الاسم الذي
تستخدمه برمجيا للوصول الى خصائصها، طرقها وحتى احداثه ا. وآنصيحة لا تحاول
ابدا تغيير اسم الأداة بعد كتابة الكثير من الآواد، فان ذلك سيضطرك الى تغيير
جميع الآواد للاسم السابق للأداة . نصيحة اخرى، لا تحاول الاعت ماد على الاسماء
Form1، Form2عند بداية رسم الأداة آ ، Visual Basic الافتراضية التي يوفرها
.... الخ، فكثرة هذه الاسماء تسبب لك تشويش على ذاآرتك. آذلك Label1
لاتستطيع اختيار اسم للأداة مادام لا يحقق الشروط التالية:
- لا يبدأ برقم.
حرف. 40 - لا يزيد عن
، ؟، " .... الخ.& - لا يحتوي على مسافة او علامات آ
- لا يكون محجوز لاسم أداة اخرى في نفس النموذج او اسم نموذج اخر في
باستثناء مصفوفة الادوات آما سيأتي بيانه لاحقا.–نفس المشروع

خصائص الموقع والحجم

أي تحتوي على–خصائص الموقع والحجم موجودة في جميع الادوات القابلة للظهور
فمن البديهي ان تكون Timer ، اما الادوات الاخرى آأداة المؤقت Visible الخاصية
هذه الخصائص غير موجودة طالما ان الأداة غير قابلة للظهور ابدا وقت التنفيذ.
تحددان موقع الزاوية العلوية اليسرى للأداة بالنسبة Top و Left خصائص الموقع
الى الأداة الحاضنة لها او موقع الزاوية العلوية اليسرى لنافذة النموذج بالنسبة الى
الشاشة. الوحدة المستخدمة هي نفس الوحدة المحددة في الخاصية
التابعة للأداة الحاضنة . اما نافذة النموذج، فدائما تكون الوحدة ScaleMode
20
آما سنعرف لاحق ا. اذا آنت تريد توسيط Twip المستخدمة لتحديد موقعها هي ال
الأداة وقت التصميم في نافذة النموذج، حدد الأداة ثم اختر احد الاوامر الموجودة
، اما في وقت التنفيذ فهذا Format من قائمة Center in Form في القائمة الفرعية
الكود يفي بالغرض:
Command1.Left) =Me.ScaleWidth -Command1.Width / (2

Command1.Top) =Me.ScaleHeight -Command1.Height / (2
فهي تمثل عرض وطول الأداة بنفس Width و Height بالنسبة لخصائص الحجم
الوحدة المستخدمة لخصائص الموقع . في حالات معينة لن تستطيع تغيير قيمة
Style اذا أتت خاصيتها– ComboBox لبعض الادوات آأداة ال Height الخاصية
هن ا ستكون معتمدة على نوع وحجم الخط Height ، فقيمة الخاصية0 نساوي
- فإن ListBox آأداة–التابعة لها. وبعض الادوات Font المستخدم في الخاصية
ليس دقيق تماما، فقيمة هذه الخاصية تمثل عدد Height التحكم في خاصيتها
ارتفاع آل سطر، فلا تتوقع ظهور جزء من سطر لان سطور النصوص x السطور
الموجودة فيها اما أن تعرض آاملة او لا تعرض.

خصائص الاحتضان

الى نافذة Reference تمثلان مرجع Container و Parent خصائص الاحتضان
Containerاو الأداة الحاضنة للأداة . Parent النموذج الحاضنة للأداة في الخاصية
Container أي لا يمكنك تعديلها، اما الخاصية–هي للقراءة فقط Parent الخاصية
فهي قابلة للتعديل في أي وقت تريد تغيير الأداة الحاضنة للأداة:
PictureBox `ادخال زر الاوامر داخل
Set Command1.Container =Picture1
في الكود السابق، لانك تقوم Set **ملاحظة:** لابد من استخدام المعامل
باسناد قيم لكائنات وليس قيم عادية، الفصل الخامس سيوضح لك
بمشيئة الله. Set اسباب استخدام المعامل
–من المرونة التي توفرها لك هاتان الخاصيتان هي امكانية الوصول الى اعضاء
خصائص وطرق - الأداة او نافذة النموذج الحاضنة للأداة، فالكود التالي يقوم بتغيير
لنافذة النموذج الحاضنة للأداة: Caption الخاصية
21
Command1.Parent.Caption"تغير عنوان النافذة" =
اذا أتت الأداة موجودة على نافذة النموذج ولم تحتضنها أداة اخرى، فان الخاصيتان
تمثلان نفس المرجع لنافذة النموذج. Container و Parent

Fontخاصية الخط

جميع الادوات التي تعرض نصوص على جبهتها، تحتوي على هذه الخاصية والتي
تحدد فيها نوع وحجم الخط المعروض على جبهة الأداة . الوقت المفضل لتحديد هذه
. اما في Font القيمة هو وقت التصميم وذلك بسبب وجود صندوق الحوار المألوف
Fontوقت التنفيذ، فعليك استخدام الكائن :

```
With Label1
.Font.Name ="Tahoma"
.Font.Bold =True
.Font.Size =20
End With
```

هي فكرة نسخ جميع خصائص الخط من Font من المرونة التي يوفرها لك الكائن
أداة الى اخرى:
Set Label1.Font =Label2.Font
التابع للأداة Fontمن المهم ان تعلم انه في الكود السابق قمنا بنسخ الكائن
اصبح أسناد واح د ترتبط Font فالكائن Label1 التابع للأداة Font مكان الكائن Label2
فان Label2 به الاداتين، والدليل انه لو قمت بتعديل احد خصائص الخط للأداة
ستتأثر ايضا: Label1 خاصية الخط التابعة للأداة
Label1 `ستتأثر ايضا الأداة
Label2.Font.Size =20
الغريبة بعض الشئ. Font السبب في التصرف السابق خاص بطبيعة الكائن
22
ملاحظة: الفصل الخامس يحتوي على العديد من الامثلة العملية، ويعرض

لتقنيات متقدمة في التعامل مع الكائنات بما في ذلك عملية نسخ
الكائنات.
فهي تمثل اسم الخط المراد استخدامه، وفي حال Font.Name بالنسبة للخاصية
يضع خط Visual Basic استخدامك لخط غير موجود في نظام التشغيل فان
افتراضي من عنده، وللاسف لا استطيع ان اذآر بالتحديد ما هو الخط الذي
فهو يختلف من جهاز الى اخر . اما بالنسبة للخطوط Visual Basic سيستخدمة
لا يتوفر بكل Font.Size ، فان حجم الخط True Typeالتي لا تندرج تحت تقنية
Visual Basic المقاسات، فلا تثق بالقيمة التي ارسلتها الى هذه الخ اصية، لان
يقوم بتغييرها:
Label2.Font.Name " =MS SystemEx"
Label2.Font.Size =20
Print Label2.Font.Size ‘ 15 الحجم
المخضرمين فالخصائص القديمة Visual Basic اخيرا، بالنسبة لمبرمجي
مع ذلك، VB6 .... الخ، ما زالت مدعومة في FontName، FontSize، FontBold
الكتاب لا يناقش خصائص غمسها التراب.

خصائص اللون

تمثلان لون الخلفية ولون الخط للأداة . بعض ForeColor و BackColor الخاصيتان
لا تدعم هاتان الخاصيتان، فألوانها تكون قياسية مستوحاه من ScorllBar الادوات آ
لا يمكن ان تلحظ التغيير في قيمة الخاصية Label نظام التشغيل . بعض الادوات آ
. آذلك الحال 1-Opaque تساوي BackStyle الا اذا أتت قيمة الخاصية BackColor
، فلن تتمكن من مشاهدة التغيير اللوني لخلفيته CommandButton مع زر الاوامر
مع العلم ان الخاصية 1–Graphical الى Style الا اذا حولت قيمة خاصيته
ليست مدعومة فيه. ForeColor
بالنسبة لقيم الالوان فمن الافضل ان اقسمها لك الى قسمين : الالوان القياسية
. الالوان القياسية هي المفضلة Custom Color والالوان الخاصة Standard Color
في معظم الاحوال للادوات لانها الوان يحددها المستخدم عن طريق خانة التبويب
من لوحة التحكم، فهي الوان Display Properties في صندوق الحوار Appearance
بهذه الالوان . Windows تناسب مزاج المستخدم ويريد ان تظهر جميع تطبيقات
لذلك، من اساليب احترام ذوق المستخدم هو استخدام الالوان التي تناسبه عن
23
. اما النوع الاخر وهو الال وان Standard Color طريق الاعتماد على الالوان القياسية
الخاصة، فهي الوان ستاتيكية أي ثابتة لا تتغير مهما قام المستخدم بتعديل
خصائص سطح مكتبه، وان لم تكن مصمم راقي في اختيار الالوان المناسبة
لادواتك، فانصحك بالانتقال الى فقرة "خصائص الجدولة ". اما اذا آنت من
المعاشرين لليوناردو دافنشي او مايكل انجلو، فتستطيع استخدام مجموعة من
الثوابت تمثل ارقام الالوان:
Me.BackColor =vbGreen
Me.BackColor =vbBlue
واستخدام نفس الاعداد التي آنت تستخدمها في ايام طفولتك QBColor او دالة
MS-DOSمع بيئة :
Me.BackColor =QBColor)0 (‘اسود
Me.ForeColor =QBColor)15 (‘ابيض
مع تحديد العمق اللوني للاحمر، الاخضر والازرق: RGB او دالة
Me.BackColor =RGB)255, 0, 0(
مليون لون، فتستطيع 16 وإن آنت تتمتع بذاآرة قوية جدا جدا جدا تحفظ آآثر من
الاستفادة من هذه الذاآرة قبل الجنون واستخدم الثوابت العددية مباشرة:
Me.BackColor =4234232
Me.ForeColor =&H53FF2 ‘قيمة ست عشرية

آل الطرق السابقة تعمل بشكل جيد في وقت التنفيذ وحتى وقت التصميم،
في خانة كتابة قيمة خصائص RGB و QBColor فتستطيع استخدام الدوال السابقة
الالوان في جدول نافذة الخصائص، رغم انها توفر لك لوح الوان يعطيك فكرة عن
عمق اللون قبل اختياره.

خصائص الجدولة

TABيفضلون استخدام مفتاح الجدولة [ ] Windows معظم مستخدمي تطبيقات
–عليها Focus للتنقل بين الادوات . معظم الادوات التي لها قابلية انتقال الترآيز

. حدد عن طريق TabIndex و TabStop آأداة النص، تحتوي على خصائص الجدولة
ما اذا آ نت تريد جعل الترآيز ينتقل الى الأداة بمجرد ان يضغط TabStop الخاصية
TabIndex]، ورتب فهرس الترآيز عن طرق الخاصية TABالمستخدم على المفتاح [
لكل أداة، مع العلم ان ترقيم الفهرس يبدأ عادة من الصفر.
للأداة، فان False تساوي TabStop **ملاحظة:** حتى لو أتت قيمة الخاصية
المستخدم لديه فرصة اخرى لنقل الترآيز الى الأداة عن طريق
النقر عليها بزر الفأرة.

خصائص مؤشر الفأرة

تحددان الشكل المطلوب MouseIcon و MousePointer خصائص مؤشر الفأرة
مؤشر قياسي 16 MousePointer . توفر لك الخاصية Mouse Cursor لمؤشر الفأرة
99يوفرها نظام التشغيل، وإن رغبت في تخص يص رمز معين، فاختر القيمة -
MouseIcon من الخاصية السابقة مع تحميل ملف المؤشر في الخاصية Custom
وقت التصميم او اأكتب الكود التالي لاجراء العملية وقت التنفيذ:

```
Command1.MousePointer =vbCustom
Command1.MouseIcon =LoadPicture )“C:\Test.ICO”(
```

لن تلاحظ تغيير المؤشر الا اذا مرر المستخدم مؤشر الفأرة فوق الأداة . مع ذلك،
من تغيير شكل المؤشر ان تم تغيير المؤشر Visual Basic هناك عدة عوامل تمنع
، جرب هذا الكود للحظة: Screen العام للبرنامج والمتمثل في الكائن

```
Screen.MousePointer =2
Command1.MousePointer =5 'لن يتغير شكل المؤشر ابدا
```

تجاهل تخصيصنا وأننا لا Visual Basic رغم اننا خصصنا رمز معين لزر الاوامر، الا ان
ليس انقاص في تقديرنا او احترامنا، وانما Visual Basic نعنيه شيئا، التجاهل من
والذي يكون آالتالي: MousePointer مع خاصية Visual Basic في اسلوب تعامل
، 0-Default غير Screen التابعة للكائن MousePointer - اذا أتت قي مة الخاصية
التابعة لسائر MousePointer سيتجاهل جميع خصائص Visual Basic فان
الادوات في البرنامج، وسيكون شكل المؤشر هو الشكل الذي تحدده في هذه
الخاصية دائما وابدا الا في حالة انتقال المؤشر الى برنامج اخر.

0-Default تساوي Screen التابعة للكائن MousePointer - اذا أتت الخاصية
، فان شكل 0-Default التابعة للأداة لا تساوي MousePointer وأتت الخاصية
التابعة للأداة. MousePointer المؤشر سيكون آما هو مطلوب في الخاصية
0-Default تساوي Screen التابعة للكائن MousePointer - اما اذا أتت الخاصية
ايضا، فان شكل 0-Default التابعة للأداة تساوي MousePointer والخاصية
التابعة لنافذة MousePointer المؤشر سيكون آما هو مطلوب في الخاصية
النموذج.
لا تقم بتغيير شكل المؤشر الا عند الحاجة لتغييره، آتحويله الى صورة يد عند
المرور فوق رابط لموقع على الانترنت، او على شكل الاسهم في حالة التحجيم،
ومن المستحسن تحويله الى شكل ساعة رملية عند بداية آل اجراء حتى يعلم
المستخدم ان عليه الانتظار:

```
Private Sub Command1_Click()
```

```
'اجراء تنفيذه يستغرق وقت
Screen.MouseIcon =vbHourglass
'آواد الاجراء
...
'لا تنسى استرجاع الشكل الافتراضي
Screen.MousePointer =vbDefault
End Sub
```

خاصية التعريب RightToLeft

ولو بسيطة - تمكنهم–، طال انتظار المبرمجين العرب لخاصية VB4 حتى VB1 منذ
VB5 من اليمين الى اليسار، وجاء–من تحويل اتجاه ادواتهم الى الاتجاه العربي
المدعومة في معظم RightToLeft حاملا البشرى السعيدة ليزف اليهم الخاصية
لا تطبق تقنية RightToLeft حتى نافذة النموذج . صحيح ان الخاصية–الادوات
Visual Basic المرآة، الا انها حلت عشرات المشاآل التي أتت تواجه مبرمجي
المخضرمين.

26

وهي تقنية تقوم Windows 98 **ملاحظة:** تقنية المرآة ظهرت من ذ الاصدار
القياسية والشائعة الى الاتجاه Windows بقلب شكل ادوات
من اليمين الى اليسار . طريقة تطبيقها تتم عن طريق–العربي
خاصة، سنتعرف عليها في الفصول API الخوض في اجراءات
اللاحقة بمشيئة الله.
وانما مكتبة MSVBVM60.DLL عليك الانتباه الى ان هذه الخاصية لا تت بع مكتبة
، الغريب في امر هذه VBAME.DLL خاصة بتطبيقات الشرق الاوسط تعرف ب
حتى تعمل معك System Directory المكتبة هو ضرورة وجودها في مجلد النظام
بشكل صحيح، فعندما تقوم بتوزيع برنامجك على اجهزة اخرى، لا تحاول وضعها في
لهذه الخاصية تحول اتجاه True نفس مجلد البرنامج فذلك لن يفيدك . القيمة
النافذة الى الاتجاه العربي آما تفعل ذلك مع اغلب الادوات.
اذا أن لديك نافذة نموذج مصممة وادواتها مرتبة بالاتجاه المعاآس للعربي، فهذا
الكود يوفر عليك عناء اعادة ترتيب الادوات لتكون من اليمين الى اليسار:

```
Private Sub Form_Load()
Dim Ctrl As Control
On Error Resume Next
For Each Ctrl In Controls
If TypeOf Ctrl Is Line Then
Ctrl.X1 =Ctrl.Container.ScaleWidth -Ctrl.X1
Ctrl.X2 =Ctrl.Container.ScaleWidth -Ctrl.X2
Else
Ctrl.Left =Ctrl.Container.ScaleWidth -Ctrl.Left -Ctrl.Width
End If
If Ctrl.Alignment =1 Then
Ctrl.Alignment =0
ElseIf Ctrl.Alignment =0 Then
Ctrl.Alignment =1
End If
Ctrl.RightToLeft =True
Next
RightToLeft =True
Err.Clear
```

27

```
End Sub
```

hWndخاصية المقبض

، وهي قيمة Read Only Propeties هذه الخاصية تعتبر من خصائص القراءة فقط
Visual Basic. حتى لو آنت من آبار المحترفين في ، Long عددية طويلة من النوع
والتي API لن تستطيع الاستفادة من هذه الخاصية الا عند تعاملك مع اجراءات
سنناقشها في الفصول اللاحقة . وبما ان الوقت مازال مبكرا جدا للحديث عنها، فاود
والادوات Standard Controls توضيح نوعين من الادوات هما الادوات القياسية
او الوهمية. Windowless Controls معدومة النوافذ
Visual Basic من صندوق الادوات، يقوم TextBox عندما تقوم بإنشاء أداة نص
طالبا منه نسخة من Windows باجراء عملية اتصالات سرية مع نظام التشغيل
يجبره على الموافقة، ويقوم باعطاء رقم فريد Windows الأداة. آرم نظام التشغيل
. هذا الرقم Window Handle لا يتكرر الى هذه الأداة يعرف بالاسم مقبض الن افذة
التابعة للأداة . من المهم ان تعلم هنا بان المسؤول hWnd يتم حفظه في الخاصية
، فجميع Visual Basic الاول والاخير عن هذه الأداة هو نظام التشغيل وليس
العمليات التنسيقية او التي يقوم بها المستخدم يتفاعل معها نظام التشغيل
هنا اشبه بالمترجم بين المبرمج وبين Visual Basic ، فدور Visual Basic وليس
Windowsنظام التشغيل .
فهي ادوات وهمية خاصة ببرامجك Windowless Controls اما الادوات من النوع
ونظام التشغيل لا يعلم أي شئ عنها مثل الاعمى، Visual Basic المصممة تحت
hWndوالدليل انها لا تمتلك الخاصية .
جميع الادوات الموجودة في صندوق الادوات هي من النوع الاول باستثناء الادوات :
فهي ادوات وهمية ولا تحتوي على الخاصية Image و Label، Timer، Shape، Line
. حاول الآثار من هذا النوع من الادوات فهي تستهلك القليل من مصادر hWnd
وتكون اسرع بكثير من الادوات الاخرى. System Resources النظام
Standard Controls ، فمعظمها من النوع ActiveX Controlsبالنسبة لادوات التحكم
. ولا يمكنك معرفة نوع أداة التحكم عن Windowless Controls وقد تكون من النوع
بها، فقد تكون الأداة من النوع الاول ولكن مصمم hWnd طريق اختبار وجود الخاصية
لاسباب شخصية. hWnd الأداة قد اخفى ظهور الخاصية

خصائص اخرى

التي تمكنك من اخفاء الأداة Visible من الخصائص المشترآة الاخرى خاصية الرؤية
والادوات المحضونة بها عن عين المستخدم لكنها ظاهرة لعين المبرمج، فالأداة
28
موجودة في الذاآرة وبامكان المبرمج الوصول اليها حتى وان أتت مخفية . خاصية
تمنع المستخدم من التفاعل مع الأداة سواء بالنقر او الكتابة Enabled التمكين
تحفظ قيمة حرفية Tag عليها وهي تؤثر ايضا على الادوات المحضونة به ا. الخاصية
تكون قيمة اضافية -لا راحت ولا جت- ولا تؤثر باي شكل من الاشكال على String
تستخدم في حالة نسخ الادوات لتكوين ما Index سلوك او مظهر الأداة . الخاصية
والذي سنتطرق اليه في فصل "الاستخدام Control Array يعرف بمصفوفة الادوات
المتقدم للنماذج".
ومن الخصائص التي تؤثر على مظهر الأداة والمدعومة على بعض الادوات خاصية
للأداة، والخاصية 3-D التي تعطي مظهر ثلاثي الابعاد Appearance المظهر
التي تخفي او تظهر الحدود الخارجية للأداة وأيضا خاصية المحاذاة BorderStyle
التي تحاذي الأداة تلقائيا حتى مع تغيير حجم النافذة دون الحاجة الى كتابة Align
آاواد اضافية.
ظهور مربع اصفر عندما تقوم بتوجيه مؤشر Windows تلاحظ في معظم تطبيقات
الفأرة الى أداة معينة والانتظار بضع ثواني دون تحريك المؤشر، هذا المربع يدعى
، بامكانك تخصيص تليمح لكل أداة موجودة في نافذة النموذج ToolTip أداة التلميح
ToolTipTextعن طريق الخاصية .

او DragIcon و DragMode اخيرا، الخصائص القديمة آخصائص السحب والالقاء .... الخ من الخصائص LinkTopic، LinkMode، LinkItem خصائص الر بط الديناميكي التي قد جار عليها الزمن وطغت عليها تقنيات افضل منه ا. اذا آنت مضطر لتحقيق مع برامجك القديمة، فهي مازالت مدعومة بشكل جيد Compatibility التوافقية جدا، اما هذا الكتاب فلن ينظر الى الخلف ابدا ولن يذآر هذه الخصائص بعد النقطة التالية.

الطرق المشترآة

بعد الخصائص تأتي الطرق، الطرق عبارة عن اوامر ترسلها الى الأداة لتحريكها او تعود Functions هي عبارة عن دوال Methods نقل التراآيز إليه ا. والواقع ان الطرق تقوم بوظيفة ما ولكنها لا تعيد أي قيمة . وآما توجد Sub's بقيم معينه، او اجراءات العديد من الخصائص المشترآة بين الادوات، توجد ايضاً عدة طرق مشترآة هي:

الطريقةMove

، فان Width و Left، Top، Height اذا أتت الأداة تدعم خصائص الموقع والحجم مدعومة بها لا محالة. فالكود التالي: Move الطريقة


```
Form1.Left  =100
Form1.Top  =200
Form1.Height  =300
Form1.Width  =400
```

سطور مملة 4 مرات الى جانب انه يستهلك Form_Resize 4 يقوم يتفجير الحدث Moveتؤدي الى بطء في التنفيذ، من هنا تبرز ميزة الطريقة :

```
Form1.Move 100, 200, 300, 400
```

جميع القيم المرسلة اختيارية باستثناء القيمة الاولى، و لا تستطيع ارسال قيمة دون ارسال قيمة سابقة لها:

```
Form1.Move 100 , 200 'ممكن عمل ذلك
Form1.Move 100 , , 300 'انسى هذه الفكرة
```

الطريقةSetFocus

الخاصة به ا. اذا أتت SetFocus توجيه التراآيز الى الأداة يتم باستدعاء الطريقة الأداة مخفية او غير ممكنة، فان هذه الطر يقة ستتسبب في وقوع خطأ وقت Visible . لذلك، ينصح بالتحقق من خاصيتي الظهور Run Time Error التشغيل قبل نقل التراآيز الى الأداة: Enabled والتمكين

```
If Text1.Visible  =True And Text1.Enabled  =True Then
Text1.SetFocus
End If
```

اذا آنت تريد منع المستخدم من نقل التراآيز الى أي أداة اخرى قبل تحقق شرط LostFocusمعين، فافضل مكان هو الحدث :

```
Private Sub Text1_LostFocus()
If Trim)Text1.Text "" = (Then
Text1.SetFocus
End If
End Sub
```


اعيد وآرر، لا تحاول استخدام هذه الطريقة الا في حالة ظهور الأداة، فلو مثلا، عليك اظهار النافذة قبل استخدام Form_Load استخدمتها في الحدث الطريقة:

```
Private Sub Form_Load()
Me.Show
Text1.SetFocus
```

End Sub

الطريقةZOrder

قد تحتاج الى اعادة وضع أداة فوق الادوات او خلف الادوات وقت التنفيذ، الطريقة تفي بالغرض لوضع الأداة فوق الادوات الاخرى، وقد تجعلها خلف الادوات ZOrder
:1 اخرى في حالة ارسال القيمة
` 0 القيمة الافتراضية
فوق جميع الادوات` Command1.Zorder
فوق جميع الادوات` Command1.Zorder 0
خلف جميع الادوات` Command1.ZOrder 1
- فانه Label آأداة العنوان– Winodwless Controls بالنسبة للادوات معدومة النوافذ . تستطيع Standard Controls من عاشر المستحيلات ان تظهر فوق أداة قياسية ان تفترض ان للنافذة طبقتين، الاولى خاصة للادوات القياسية والثانية خاصة للادوات معدومة النوافذ والتي تكون خلف الطبقة الاولى دائم ا. آذلك، الادوات الحاضنة تكون خلف الادوات المحضونة بها. وبالنسبة لنوافذ النماذج، فيمكنك استخدام هذه الطريقة لوضع نافذة نموذج فوق النوافذ الاخرى او خلفها، ولكن لا بصورة دائمة Windows يمكنك جعل نافذة النموذج في مقدمة نوافذ جميع تطبيقات باستخدام هذه الطريقة.

الطريقةRefresh

هذه الطريقة تطلب من الأداة اعادة رسم نفسها. عملياً لن تحتاج لاستدعاء هذه يقوم برسم الأداة تلقائيا بمجرد تغيير قيم خصائصه ا. Visual Basic الطريقة آثيرا ف لإعادة رسم الأداة في حالات الضغط الشديد عليه: Visual Basic الا انك قد تجبر
Private Sub Command1_Click()
31
Dim X As Long
For X =0 To 100000
Label1.Caption =CStr)X(
Label1.Refresh
Next
End Sub
. في DoEvents القدماء باستخدام الدالة Visual Basic قد يقترح علي احد مبرمجي البداية ساشكره على اقتراحه الذآي ولكن سأرفض اقتراحه هنا لان وظيفة هذه الدالة ليست مقصورة على اعادة الرسم فقط وانما تتعدى هذا الدور بكثير، فهي لباقي اجزاء البرنامج وليس الادوات فقط، Processing خاصة لعملية توزيع المعالجة مما يؤدي الى بطئ في السرعة . ليس هذا فقط، بل قد تؤدي الى شوائب برمجية Command1 ، فهي تعطي فرصة آبيرة للمستخدم لاعادة الضغط على الزر Bugs مرة اخرى قبل ان ينتهي الاجراء من آامال الحلقة التكرارية الاولى . على العموم، شكرا على الاقتراح!

الاحداث المشترآة

تقتضي عملية Event Driven Programming فلسفة البرمجة المسيرة بالاحداث تنفيذ الاآواد عند حالات معينة تعرف بوقوع الاحداث او انفجار الاحداث. فعندما تصلك رسالة امر من رئيسك في العمل، فإن استجابتك للحدث تكون بتنفيذ ما يطلب منك. آذلك الحال مع الادوات، فالاآواد التي تضعها لن يتم تنفيذها الا عند وقوع اسمائها تتبع الصيغة: Sub's الحدث عليها. والاحداث عبارة عن اجراءات
الحدث_اسم الكائن
Form_Click ()
Command1_Click ()
عوضا عن التعبير Fire Event **ملاحظة:** استخدم التعبير تفجير الحدث
استدعاء الحدث، فاستدعاء الحدث هي عملية كتابة اسم الحدث

لتنفيذه آما تفعل مع الاجراءات، اما تفجير الحدث فهي عملية
، فأرجو ان لا Visual Basic استدعاء الحدث من قبل نظام التشغيل و
تتعجب من آثرة استخدامي لهذا المصطلح حتى نزول اسمي في
القائمة الامريكية للمشتبه فيهم بالارهاب!

وليس اسم Form بالنسبة لنافذة النموذج، تسمية احداثها دائما ما تبدأ بالكلمة
. وآما علمنا بوجود خصائص وطرق مشترآة Name النموذج الموجود في الخاصية
بين الادوات، فان الاحداث لا تشذ عن هذه القاعدة:

احداث الفأرة

من آاودك المستجابة تكون ردة فعل لاعمال درامية قام بها المستخدم 50%
بالفأرة. اول حدث تعرضه لك معظم الادوات عند النقر المزدوج عليها هو الحدث
DblClick والذي ينفجر في لحظة النقر على الأداة بزر الفأرة الايسر . والحدث Click
وتعتقد انه لا ينفجر الا في حالة Click يمثل النقر المزدوج . لا تثق آثيرا في الحدث
CheckBox للاداتين Value النقر بزر الفأرة الايسر، فعند قيامك بتغيير قيمة الخاصية
التابع للأداة. نفس Click تلقائيا بتفجير الحدث Visual Basic ، يقوم OptionButton و
و ListBox التابعة للاداتين ListIndex الانفجار يحدث عندما تقوم بتغيير الخاصية
.ComboBox

من الاساليب الخاطئة التي يتبعها قليل من المبرمجين هي كتابة آاواد في آلا
Visual لنفس الأداة، رغم انك تستطيع عمل ذلك ب DblClick و Click الحدثين
، الا انها طريقة غير مرنة تسبب التشويش على مستخدم برنامجك تؤدي به Basic
الى الاستغناء عن الفأرة . فلو قام المستخدم بالنقر المزدوج على الأداة، فان
. اذا أن لابد من DblClick سيتم تنفيذه اولا ومن ثم تنفيذ الحدث Click الحدث
استخدام الحدثين في أداة واحدة، فاتمنى من صميم قلبي معرفة الحدث المقصود
قبل تنفيذه حتى لا يستغني المستخدم عن فأرته:

```
Dim bDblClick As Boolean ‘متغير عام
Private Sub Form_Click()
Dim X As Single
bDblClick =False
‘اعطاء مهلة نصف ثانية
X =Timer
Do
DoEvents
If bDblClick Then Exit Sub
Loop Until Timer > X +0.5
‘أكتب الآواد هنا
‘...
```



```
End Sub
Private Sub Form_DblClick()
bDblClick =True
‘أكتب الآواد هنا
‘...
End Sub
```

اذا آنت تريد معرفة المزيد من التفاصيل حول عملية النقر التي قام بها المستخدم،
آموقع مؤشر الفأرة او الزر الذي استخدمه المستخدم سواء الايمن او الايسر ....
MouseDownالخ من تفاصيل دقيقة، فيسرني ان اعرض عليك الاحداث ،
التي تعطيك تفاصيل آاثر عن عمليات الفأرة على شكل MouseUp و MouseMove
Shift، المفاتيح المضغوطة ، Button متغيرات مرسلة هي: نوع الزر المستخدم
Yوالاحداثي الصادي للمؤشر . X الاحداثي السيني للمؤشر

، فقد يكون الزر الايمن و /او الايسر و /او الاوسط Button بالنسبة للزر المستخدم
للفأرة، هذا المثال يعطيك فكرة عن طريقة معرفة الازرار المضغوطة:

```
Private Sub Form_MouseDown)Button As Integer, Shift As Integer, _
X As Single, Y As Single(
If Button And vbLeftButton Then
' الزر الايسر مضغوط
End If
If Button And vbRightButton Then
' الزر الايمن مضغوط
End If
If Button And vbMiddleButton Then
' الزر الاوسط مضغوط
End If
End Sub
```

الخاص باحداث لوحة المفاتيح Shift ، فهو نفس المتغير Shift اما المفتاح المضغوط
والمثال التطبيقي للتعامل معه موجود في فقرة "احداث لوحة KeyDown و KeyUp
فهي تمثل موقع Y و X المفاتيح" التي ستصل اليها قريب ا. وبالنسبة للاحداثيات
)0 ،0مؤشر الفأرة بالنسبة للأداة نفسها وليس الشاشة، حيث تمثل النقطة (
34
آلما اتجه مؤشر الفأرة الى جهة X الزاوية العلوية اليسرى للأداة، وتزداد قيمة
آلما اتجه مؤشر الفأرة الى الاسفل. Y اليمين وتزداد قيمة
بمجرد ان يقوم المستخدم بتحريك المؤشر فوق MouseMove يتم تفجير الحدث
الأداة، ونهاية الحدث تكون لحظة خروج المؤشر عن حدود الأداة . اما في حالة
سيتم تفجيره حتى لو تعدى المؤشر MouseMove فان الحدث Capturing الالتقاط
في حالة آون مؤشر الفأرة Y و X حدود الأداة مما يترتب عنه قيم سالبة للاحداثيات
زحف يسار او فوق الأداة.
هي عملية الضغط بزر الفأرة Capturing **ملاحظة:** المقصد من آلمة الالتقاط
على الأداة مع استمرار الضغط على الزر.
فسيتم تفجيرهما بمجرد الضغط على MouseUp و MouseDown بالنسبة للحدثين
زر الفأرة و تحرير الزر على التوالي حتى لو اختلفت الازرار، فلو قمت بالضغط على
وأبقيته مضغوط ا- ومن ثم قمت بالضغط على زر الفأرة الايمن،–زر الفأرة الايسر
مرتين، وعند تحرير الازرار، فان MouseDown بتفجير الحدث Visual Basic فسيقوم
مقبلان. MouseUp انفجارين للحدث
على الأداة، فان Double Click اخيرا، في حالة قيام المستخدم بالنقر المزدوج
ترتيب وقوع او انفجار الاحداث يتم على النحو التالي:
MouseUp < -DblClick < -MouseMove < -Click < -MouseUp < -MouseDown
- مرة اخرى.> MouseMove

احداث الترآيز

LostFocus عندما تستقبل الأداة الترآيز، والحدث GotFocus يتم تفجير الحدث
عندما تفقد الأداة الترآيز، سواء أن ذلك بالفأرة او لوحة المفاتيح أو برمجي ا. أما
بالنسبة لنافذة النموذج، فهذه الاحداث تعمل جيدا بها شريطة عدم وجود أي أداة
قابلة لاستقبال الترآيز.
ملاحظة: لن تتم عملية تفجير الاحداث بالطريقة المتوقعة اذا فقدت النافذة
ترآيزها بسبب الانتقال الى تطبيق اخر او استقبلت ترآيزها بع د
الانتقال من تطبيق اخر . باختصار، احداث الترآيز لا تعمل الا بين
نوافذ وادوات برنامجك فقط.

احداث لوحة المفاتيح

KeyPressناتجة من لوحة المفاتيح هي ، Visual Basic ثلاثة احداث مرنة يوفرها لك
. فعندما يقوم المستخدم بالضغط على أي زر من ازرار لوحة KeyUp و KeyDown
بتحويل المفتاح Visual Basic سيتم تفجيره، ثم يقوم KeyDown المفاتيح، فالحدث
، وبعد ان KeyPress ثم يتم تفجير الحدث ASCII المدخل الى مقابله في جدول
بالانفجار. KeyUpيرفع المستخدم اصبعه عن المفتاح يبدأ الحدث
في حالة قيام المستخدم الضغط Visual Basic فيفجره KeyPress بالنسبة للحدث
[والحروف ENTER]، [BACKSPACE]، [ESCAPE]، [Ctrl+...على المفاتيح]
المطبوعة، اما المفاتيح الاخرى آالاسهم او مفاتيح الوظائف وغيره ا... فلا تؤدي
لها نصيب من KeyUp و KeyDown ولكن الاحداث KeyPress الى انفجار الحدث
متمثلة Integer قيمة عددية من النوع KeyPress الوقوع. المزيد ايضا، يرسل الحدث
تمثل المقابل العددي للحرف المدخل في جدول KeyAscii في متغير عددي بالاسم
:ASCII

```
Private Sub Form_KeyPress)KeyAscii As Integer(
Print Chr$)KeyAscii (&  " = "& KeyAscii
End Sub
```

مرسل بالمرجع وليس القيمة اي يمكنك تعديل قيمته مما يترتب KeyAscii المتغير
عليه مرونة آبير في التحكم في مدخلات المستخدم، هذا الكود مثلا يحول جميع
Capitalالحروف المدخلة في أداة النص الى حروف آبيرة :

```
Private Sub Text1_KeyPress)KeyAscii As Integer(
KeyAscii  =Asc)UCase)Chr$)KeyAscii(((
End Sub
```

واذا اسندت قيمة الصفر الى هذا المتغير، فانك قد الغيت عملية ارسال قيمة
المفتاح الى الأداة المستقبلة له . هذا الكود مثلا يمنع المستخدم من كتابة أي
:9 ، ...،1 ،0 شئ في أدا النص عدا الاعداد

```
Private Sub Text1_KeyPress)KeyAscii As Integer(
If KeyAscii < Asc")0 ("Or KeyAscii > Asc")9 ("Then
KeyAscii  =0
End If
End Sub
```

36

.Chr $ و Asc **ملاحظة:** تلاحظ انني اعتمد في الامثلة السابقة على الدالتين
مع ذلك، يمكنك الاستغناء عنهما اذا آنت تعرف المقابل العددي
ASCIIللحرف المطلوب في جدول .
وتمثل المفتاح KeyCode بقيمتين الاولى KeyUp و KeyDown يزودك الحدثين
[فيما ALT] و [SHIFT]، [CTRLوتمثل حالة المفاتيح [Shift المدخل، والثانية هي
اقصد مضغوطة- او لا آما في الكود التالي:–اذا أتت مفعوصة

```
Private Sub Form_KeyDown)KeyCode As Integer, Shift As Integer(
If Shift And vbShiftMask Then
' [ مضغوط SHIFTالمفتاح ]
End If
If Shift And vbCtrlMask Then
' [ مضغوط CTRLالمفتاح ]
End If
If Shift And vbAltMask Then
' [ مضغوط ALTالمفتاح ]
End If
End Sub
```

- هي القيمة KeyCode بالنسبة الى قيمة المفتاح المدخل -المتمثلة في المتغير

، الا KeyAscii الفيزيائية للمفتاح في لوحة المفاتيح، صحيح انها مثل قيمة المتغير
او علامات آ Small Letter انها لا تمثل نوعية الحرف المدخل سواء أن صغير
.... الخ، او حتى حروف عربية ا، ب، ت .... الخ. فهي ترسل دائما الق يمة@#؟%
.... الخ . المزيد ايضا، لا يمكننا تعديل Capital آ C ،B ،A للحرف الانجليزي الكبير
آما آنا فعلنا في الصفحة السابقة مع المتغير KeyCode قيمة المفتاح المدخل
.KeyAscii
يتم تفجيرها عندما KeyUp و KeyPress ،KeyDown اخيرا، احداث لوحة المفاتيح
يكون الترآيز على الأ داة المكتوب فيها الكود، واذا وجدت احداث اضافية تابعة لنافذة
النموذج وسألتني أي الاحداث سيتم تنفيذها اولا؟، هل هي الاحداث التابعة لنافذة
النموذج ام الأداة التي عليها الترآيز؟ فساخبرك بان لديك عقلية نبيهة جدا جدا
بسببه !! فان أتت قيمة الخاصية Visual Basic تستحق ان تكون مبرمج
، فان النافذة ستفجر احداثها اولا True التابعة لنافذة النموذج تساوي KeyPreview
، فان نافذة False ومن ثم الأداة التي عليها الترآيز، اما ان أتت قيمة هذه الخاصية
37
النموذج ستتجاهل هذه الاحداث وأنها غير موجودة، ولن تفجر الا احداث الأداة
فقط.

حدث التغيير Change

بمجرد القيام بتغيير محتويات الاداة آتغير النص Change يتم تفجير حدث التغيير
. ولكن الاعتماد على هذا الحدث فيه Text او الخاصية Caption الظاهر في الخاصية
و CheckBox للأداتين Value شيء من الخطأ، فعند تغيير قيمة الخاصية
بتفجير هذا الحدث، آذلك عند تغيير الشكل Visual Basic لن يقوم OptionButton
الظاهري للادوات آحجمها او الوانها لن يتم تفجير هذا الحدث.

نافذة النموذج

، فهي البؤرة التي Visual Basic نافذة النموذج عزيزة على قلوب جميع مبرمجي
والذي ص احبني منذ Form1 من خلالها مع الاسم الابتدائي لها Visual Basic نرى
. ولحبي لها وتقديري للعشرة الطويلة بيني وبينها، VB1 عشر سنوات مع بدايات
قررت تخصيص فقرة خاصة بها في هذا الفصل وفصل آامل "الاستخدام المتقدم
للنماذج" في هذا الكتاب، عساها ان تميزني بين المبرمجين آما ميزتها عن سائر
الكائنات!
قبل ان اخوض في تفصيل ناف ذة النموذج واتحدث عن خصائصها، طرقها واحداثها،
Form Templatesاو قوالب النماذج ، Templates بودي التطرق الى فكرة القوالب
وهي عبارة عن نماذج جاهزة التصميم ومضبوطة الخصائص تستخدمها في برامجك
Add اليومية بصورة متكررة دون الحاجة الى اعادة تصميمها من الصفر . اختر الامر
وستفهم الفكرة من قوالب النماذج الجاهزة. ففي صندوق Project من قائمة Form
الحوار الذي سيظهر امامك، ستجد العديد من النماذج التي تستخدمها آثيرا في
برامجك الاعتيادية، واذا أتت لا تملأ بريق عينيك، صمم يا مصمم النماذج آما تريد،
-او المسارالمحدد في \ VB98\Template\Forms ومن ثم قم بحفظها في المجلد
- ستلاحظ وجود نافذتكOptions في صندوق الحوار Environment خانة التبويب
بين القوالب السابقة. Template آقالب

خصائص النموذج

بعد ان تبرق نافذة النموذج امام عينيك، ستبدأ بوضع الادوات عليها ومن ثم
. القيمة BorderStyle تحجيمها. وبعد ذلك، تقوم باخت يار شكل حدودها مع الخاصية
لبرامجي لانها Splash Screen لا استخدمها الا في الشاشة الافتتاحية 0-None
38
، فتمنع المستخدم من امكانيات TitleBar تخفي حدودها وحتى شريط عنوانها
تحريك النافذة وتسبب له حالة من الندم على تشغيل برنامجك ان لم يقم بالضغط
] آي يتمكن من اغلاق Ctrl+Shift+Esc] او [Ctrl+Alt+Delعلى المفاتيح [

فستمكن المستخدم من تحريك النافذة بانسيابية 2-Sizable برنامجك. اما القيمة
ان–مريحة وتمكنه ايضا من اعادة تحجيم النافذة بالشكل الذي يناسبه ولن يقوم
3-Fixed Dailog و 1-Fixed Single شاء الله- باستخدام المفاتيح السابقة. القيمت ان
تمنع المستخدم من اعادة تحجيم النافذة مع ابقاء شريط العنوان وهي قيم قياسية
، والفرق بين القيمتين يظهر جليا Windows تطبيقات Dialog Boxes لصناديق حوار
على شريط Minimize والتصغير Maximize في ان الاولى تسمح بظهور زر التكبير
5-Sizable و 4-Fixed ToolWindow العنوان اما الثانية فلا. استخدم القيمتين
لتصغير ارتفاع شريط العنوان وهي موضة لنوافذ اشرطة الادوات. ToolWindow
لكن مع حدود 0-None توجد قيمة سادسة لشكل الحد اشبه ما تكون بالقيمة
، ولن تستطيع مشاهدتها الا ان أتت القيمة الموجودة في 3-D ثلاثية الابعاد
، مع اختيار False تساوي ControlBox خالية، وقيمة الخاصية Caption الخاصية
BorderStyleمن الخاصية السابقة . 3-Fixed Dailog القيمة
تستطيع اظهار، اخفاء او تمكين صندوق التحكم او ازرار التكبير والتصغير عن طريق
. النص الذي سيظهر في شريط MinButton و ControlBox، MaxButtom الخصائص
. تستطيع توسيط Caption عنوان النافذة هو نفس النص الموجود في الخاصية
من قيم الخاصية 2-Center النافذة وسط الشاشة عن طريق اختيار القيمة
، وبامكانك منع المستخدم من تحريك النافذة عن طريق تحويل StartupPosition
فهي تضيف زر ShowInTaskBar . اما الخاصية False الى Moveable قيمة الخاصية
بحيث يتمكن Start بجانب زر "ابدأ" او Windows Task Bar الى شريط المهام
المستخدم من تنشيط نافذة برنامجك بمجرد النقر على هذا الزر . اخيرا، خاصية
التي تمكنك من تكبير النافذة لتغطي آامل الشاشة، تصغيرها او WindowState
استرجاع الحجم الطبيعي لها.

خصائص الصور:

تحدد ما اذا أتت اعادة رسم نافذة النموذج تتم AutoRedraw عن طريق الخاصية
بواسطة آاوادك . في الحالة الاولى فان سرعة اعادة False او يدويا True تلقائيا
الرسم تكون اسرع من الحالة الثانية، الا انها تستهلك الاف الكيلوبايت ات في الذاآرة
المتاحة للبرامج الاخرى، لك ان System Resources مما يقلل من مصادر النظام
1406 تستهلك True Color مع عمق لوني 800x600 تتخيل نافذة حجمها
نوافذ جشعة من برنامجك 9 او 5 ميجا، ولك ان تتخيل1 ما يزيد على–آيلوبايت
مفتوحة، وآم ستسبب من البطء في التحميل والحجز الكبير في الذاآرة؟ من
39
لن يتم تفجيره ابدا طالما أتت قيمة Form_Paint المهم ان اذآر هنا بان الحدث
. باختصار، لا تستخدم هذه الخاصية الا عند الحاجة الماسة True هذه الخاصية
.... الخ بين سطور الحدث Print، Line فقط، وحاول وضع آاواد الكتابة والرسم آ
.Form_Paint
.... الخ بكثرة، فانصحك بتغيير قيمة Line، Circle اذا آنت تستخدم طرق الرسم
حتى تزيد سرعة طرق الرسم بمقدار الضعف لان False الى ClipControls الخاصية
ولن يقوم باعادة الرسم الا Clipping region لن يقوم بإنشاء منطقة Visual Basic
للمناطق التي تحتاج الى اعادة رسم، اما اذا لم تستخدم طرق الرسم، فالقيمة
تكون مناسبة لهذه الخاصية. True
Device Context تخيرك فيما لو آنت تريد إنشاء سياق رسم HasDC الخاصية
لنافذة النموذج ام لا، سياق الرسم عبارة عن ترآيب خاص بنظام التشغيل يحمل
، فاجعل Picture مواصفات وبيانات الصورة . اذا آنت لا تنوي وضع صورة في الخاصية
آي تقلل من استهلاك مصادر النظام مع العلم ان False قيمة هذه الخاصية
لن تعمل معك الا اذا قمت بتحميل صورة على نافذة النموذج hDC الخاصية
قيمة مؤقتة تزول مباشرة بعد زوال الصورة. hDC فستحمل الخاصية
التابع لنافذة Control Box تمثل الرمز الذي يظهر في صندوق التحكم Icon الخاصية
النموذج والرمز الظاهر على زر النافذة في شريط المهام ، هذا إذا أتت الخاصية

) فان نظام None، اما ان أتت قيمة الخاصية (True تساوي ShowInTaskbar
تساوي ControlBox التشغيل يضع رمز افتراضي شريطة أن تكون قيمة الخاصية
EXE File Icon . من الضروري أن تعلم انه لا يمكنك تخصيص رمز البرنامج True
يخيرك بين احد رموز نوافذ النماذج التابعة Visual Basic بشكل مستقل، ف
الموجودة في صندوق Make من خانة التبويب Icon لمشروعك عن طريق القائمة
Project Propertiesحوار خصائص المشروع .
تمكنك من تحميل ملف صورة ووضعه في داخل نافذة النموذج، Picture الخاصية
JPG ،GIF ،DIB ،BMPتدعم هذه الخاصية هيئات مختلفة من الملفات هي : ،
. تستطيع تحميل ملف الصورة وقت التصميم باختيار اسم CUR و WMF، EMF، ICO
الملف من صندوق حوار الخاصية، او استخدام طريقة اخرى افضلها آثيرا و هي
ومن ثم Clipboard الصورة من البرنامج الذي يعرضها الى الحافظة Copy نسخ
. واذا اردت وضع الصورة في وقت التنفيذ، Edit من القائمة Paste لصقها باختيار الامر
تمكنك من فعل ذلك او سرقة صورة تابعة لكائن آخر: LoadPicture فالدالة
Form1.Picture =LoadPicture)"C:\Turki.BMP " ('تحميل صورة وجهي الوسيم!
Form2.Picture =Form1.Picture ' Form1 نفس الصورة الموجودة في النموذج

40

هي أسناد يحتوي على خصائص اضافية آعرض Picture **ملاحظة:** الخاصية
الصورة وارتفاع وغيرها:
Print Me.Picture.Height
Print Me.Picture.Width
Icon تمكنك من استخلاص رمز LoadPicture متغيرات جد يدة الى الدالة VB6 اضاف
MSDNتجد شرح وافي لها في مكتبة ، ICO من مجموعة رموز مضمنة في ملف
التي SavePicture ، فما المانع من ذآر زميلتها LoadPicture وبما انني ذآرت الدالة
تمكنك من حفظ الصورة الى ملف:
SavePicture Form1.Picture, "C:\Aseeri.BMP"
هي نفس هيئة SavePicture التي تحفظ بها الدالة Format**ملاحظة:** الهيئة
فيتم JPG و GIF الصورة التي حملت في الخاصية، باستثناء الهيئات
BMPتحويلهما الى الهيئة .

خصائص الرسم:

تمثل الرسمة الموجودة على نافذة النموذج الناتجة من استخدام Image الخاصية
.... الخ، وستكون دائما فوق الصورة الموجودة في الخاصية Line، Cirle طرق الرسم
ان صح التعبير - من هذه الخاصية الا–، لن تستطيع استخدام او الاستفادة Picture
Trueهي . AutoRedraw ان أتت قيمة الخاصية
تحدد عرض او سمك الفرشاة المستخدمة لرسم الخطوط DrawWidth الخاصية
فهي تحدد ForeColor اما الخاصية Circle و PSet، Line والاشكال بطرق الرسم
فهي تمكنك من DrawStyle اللون الافتراضي للطرق السابقة . بالنسبة للخاصية
Cirlceو ، Line تحديد شكل النقش لرسم الخطوط والدوائر باستخدام الطريقتين
للمنطقة الداخل ية من المربع او الدائرة مع لون FillStyle آذلك تمكنك منه الخاصية
CurrentY و CurrentX . اما الخاصيتان FillColor التعبئة الموجود في الخاصية
Print، Line فتمثلان الاحداثيات الحالية التي تستخدم لطرق المخرجات والرسم
.... الخ، واللتان تتأثران بكل عملية رسم او خرج باستخدام الطرق السابقة . اما
Printفهي تحدد اسلوب خرج الطباعة باستخدام الامر ، FontTransparent الخاصية
فسيكون لون خلفية الطباعة هو نفس لون False فإن أتت قيمة الخاصية تساوي
فإن خلفية الطباعة True للنموذج، أما إن أتت قيمة الخاصية BackColor الخلفية
ستكون شفافة.

41

DrawModeالخاصية :

من اقوى خصائص نافذة النموذج الرسومية، فعن طريقها DrawMode تعتبر الخاصي ة

- مع النقاط Line آ–تحدد طريقة التفاعل بين الرسوم التي ترسمها بطرق الرسم
13-Copy Pen الموجودة على نافذة النموذج. القيمة الافتراضية لهذه الخاصية هي
وتعني ان اللون سيظهر آما هو مطلوب، فالمربع الابيض سيكون ابيض ولو رسم
على مربع اسود، والدائرة الحمراء سترسم حمراء ولو على سطح ارجواني . الا انك
في بعض الحالات الفنية تود ان ترسم رسوم تتأثر بالالوان الموجودة على لوحة
:1- 2الرسم وهذا مثال واقعي تجده آثيرا في برامج الترآيب **شكل**
(ا)
(ب)
(ج)
على مخرجات الرسم. DrawMode **:** تأثير الخاصية
(ا) ان المستطيل الازرق الذي رسمناه على المنطقة
البيضاء قد رسم بشكل جيد جدا، ويظهر الفرق في الفن التصميمي واضحا بين
الشكلين (ب) و (ج)، ففي الشكل (ب) قمنا برسم المستطيل الازرق آما نريده
" ولن يتمكن المستخدم من رؤيته، 50ازرق مما اثر وغطى على النص المكتوب "%
بحيث DrawMode اما في الشكل (ج) فقد استخدمنا القيمة المناسبة للخاصية
تقلب اللون الازرق الى ابيض في حالة الرسم فوق اللون الاسود . لمعرفة آيف تتم
عملية تغيير الالوان، عليك ان تعلم ان الالوان في حقيقتها م ا هي الا اعداد تتحول
، اللون الذي تستخدمه يسمى 10011101010 بالنظام الثنائي الى ارقام شبيه ب
42
Screen ، واللون الموجود على لوح او سطح الرسم يسمى Pen Color لون القلم
تقوم بتطبيق المعادلة التالية: DrawMode للخاصية 15-MergePen ، فالقيمة Color
S =S Or P
واللون الموجود على الشاشة هو 170= 10101010 فلو أن ا للون المستخدم هو
سيكون: 15-MergePen من تأثير القيمة–، فان اللون الناتج 85= 01010101
S =01010101
P =10101010
S =S Or P
S =10101010 Or 01010101
S =11111111
اردت معرفة جميع المعادلات الت ابعة للقيم __________ . اذا 255 وهو 11111111
الاخرى،
DrawModeبها جدول جميل جدا تصل اليه بكتابة الجملة " MSDN فمكتبة
Index" في الفهرس . Property
ScaleModeالخاصية **:**
في بداية الفصل وبالتحديد عند فقرة "خصائص الموقع والحجم " ذآرت ان الوحدة
المستخدمة لقياس احداثيات مواقع وطول وعرض الادوات هي الوح دة الموجودة
Units قيم تمثل وحدات 8 . توفر لك هذه الخاصية ScaleMode في الخاصية
0.72 تعادل 2-Point سم، 0.567 والتي تعادل 1-Twipتستخدم للقياس هي:
120 Twips تعادل 4-Character تعادل نقطة واحدة على الشاشة، 3-Pixel انش،
تعادل ملم 6-Milimeter تعادل انش واحد، 5-Inch عاموديا، 240 Twips افقيا و
وحدة قياس خاصة يتم تعريفها من 0-User تعادل واحد سم و 7-Centimeter واحد،
قبل المبرمج.
Twipتعودان بعرض وارتفاع النافذة دائما بالوحدة ، Height و Width الخاصيتان
تؤثر على الوحدة المستخدمة في الادو ات ScaleMode فالقيمة التابعة للخاصية
المحضونة فقط وليس الحاضنة، اما لمعرفة عرض وارتفاع نافذة النموذج بوحدة غير
واستعلم عن العرض عن ScaleMode ، قم بتحديد الوحدة في الخاصية Twip ال
ScaleHeightوالارتفاع عن طريق الخاصية : ScaleWidthطريق الخاصية
Private Sub Form_Paint()

Cls
بالبكسل` ScaleMode =vbPixels
Print ScaleHeight
43
Print ScaleWidth
End Sub
تعودان بعرض وارتفاع ScaleHeight و ScaleWidth في الحقيقة، الخاصيتان
Height و Width المساحة الداخلية لنافذة النموذج، بينما تشمل الخاصيتان
المساحة الداخلية والخارجية المتمثلة في سمك حدوده ا وارتفاع شريط عنوانه ا.
مع ذلك، لن تفرق آثيرا معك فنادرا ما تحتاج المساحة الخارجية للنافذة، على
العموم هذا الكود يطبع الفرق:
Private Sub Form_Paint()
Cls
ScaleMode =vbTwips
Print Height -ScaleHeight
Print Width -ScaleWidth
End Sub
هي وحدة تعرف من قبل المبرمج، تستطيع تعريف وحدة 0-User اخيرا، القيمة
ScaleWidth، ScaleHeightخاصة بك عن طريق اسناد قيم الى الخصائص ،
. قد تحتاج تعريف وحدة قياس رسم خاصة بك في حالات ScaleTop و ScaleLeft
نادرة تعتمد على عرض المخططات الرسومية بشكل استثنائي.

طرق النموذج

ScaleModeبما ان الفقرة السابقة تحدثت عن وحدات القياس التابعة للخاصية ،
. هذه الطرق تمكنك من اجراء عملية ScaleY و ScaleX فسأبدأ بالتحدث عن الطرق
تحويل القياسات بين الوحدات السابقة افقيا وعامودي ا. ارسل القيمة ثم وحدتها
الاصلية ثم الوحدة المطلوبة:
Twips `الى Pixels التحويل من
Print ScaleX)100, vbPixels, vbTwips(
تخفيه، نستطيع ان نقول Hide تؤدي الى اظهار النموذج والطريقة Show الطريقة
ولكن على شكل طرق: Visible بكل ثقة انهما يمثلان الخاصية
Form1.Show ` Form1.Visible =True
Form1.Hide ` Form1.Visible =False
44

طرق الرسم

تمسح جميع الرسوم الموجودة على النافذة وتصفر الاحداثيات Cls الطريقة
تعود بالقيمة العددية Point)، والطريقة0 , 0الى الاحداثي (CurrentY و CurrentX
) على النافذة: x, yللون الموجود في الاحداثي (
Private Sub Form_Load()
'تحميل صورة وجهي الوسيم!
Form1.Picture =LoadPicture)"C:\Turki.BMP"(
End Sub
Private Sub Form_MouseMove)Button As Integer, Shift As Integer, _
X As Single, Y As Single(
Label1.BackColor =Point)X, Y(
Label1.Caption =Hex$)Point)X, Y((
End Sub
) للنقطة، X, Yتمكنك من رسم نقطة على النافذة بارسال الاحداثي (PSet الطريقة
او بامكانك ForeColor سيكون لون النقطة هو نفس اللون الموجود في الخاصية
ارسال اللون:
Me.ForeColor =vbBlack

PSet)0 , 0 (`نقطة سوداء
PSet)500, 500(, vbRed `نقطة حمراء
والتي تضيف Step وطرق الرسم الاخرى - تدعم الكلمة المحجوزة– PSet الطريقة
الموجودة في الخاصيتان–) الى الاحداثيات الحالية X, Y الاحداثيات المرسلة (
CurrentX : وCurrentY
Private Sub Form_Paint()
Dim X As Integer
Cls
Me.CurrentX =0
Me.CurrentY =0
For X =0 To 100

45

PSet Step)5, 5(
Next
End Sub
(والنهاية X1, Y1 تمكنك من رسم الخطوط بارسال احداثيات البداية (Line الطريقة
):X2, Y2(
ForeColor =vbGreen
Me.Line)0, 0) - (Me.ScaleWidth, Me.ScaleHeight) `خط اخضر
Me.Line)0, Me.ScaleHeight) - (Me.ScaleWidth, 0(, vbRed `خط احمر
و CurrentX) فان القيم الحالية للخصائص X1, Y1 في حالة تجاهلك للاحداثي (
هي نقطة البداية: CurrentY
Private Sub Form_MouseMove)Button As Integer, Shift As Integer, _
X As Single, Y As Single(
Me.Line) -X, Y(
End Sub
B تمكنك من رسم المستطيلات عوضا عن الخطوط بارسال الحرف Line الطريقة
FillColor واللون : FillStyle مع العلم ان النقش سيكون النقش المحدد في الخاصية
ForeColor =vbWhite
Line)0, 0) - (200, 200(, , B `مربع ابيض
Line – Step)200, 200(, vbBlue, B `مربع ازرق
BF دون استخدام الخصائص، ارسل الحرفين :–ولتلوين المربع مباشرة
ForeColor =vbWhite
Line)0, 0) - (200, 200(, , BF `مربع ابيض
Line – Step)200, 200(, vbBlue, BF `مربع ازرق
التي من الواضح انها لا ترسم نجوم ! وانما Cirlce واختم فقرة طرق الرسم بالطريقة
دوائر بارسال احداثي نقطة مرآز الدائرة ومن ثم طول قطرها:

46

ForeColor =vbWhite
Circle)Me.ScaleWidth /2, Me.ScaleHeight /2 (, 500 `دائرة بيضاء
Circle)Me.ScaleWidth /2, Me.ScaleHeight /2(, 200, vbGreen `دائرة خضراء
، وانما بالقيمة BF آما في الطريقة BF عملية تلوين الدائرة لا تتم باستخدام
FillStyle مع النقش . FillColor الموجودة في الخاصية
هي امكانية رسم الاقواس بتحديد Cirlce من الاشياء التي تعجبني في الطريقة
Radian زاوية النهاية والبداية بوحدة الراديان :
Const PI =3.14
`نصف دائرة فتحهتها الى الاعلى
Circle)Me.ScaleWidth /2, Me.ScaleHeight /2(, 1000, , 0, PI
`نصف دائرة فتحهتها الى الاسفل
Circle)Me.ScaleWidth /2, Me.ScaleHeight /2(, 800, , PI, 0
`ربع دائرة

Circle)Me.ScaleWidth /2, Me.ScaleHeight /2(, 500, , 0, PI /2
هي امكا نية ايصال اطراف الاقواس Circle من الاشياء التي تزيد من قوة الطريقة
، ولعمل ذلك Charts بمرآز الدائرة وتلوين جزء معين آما يحدث مع المخططات
يشترط استخدام القيم السالبة:
Const PI =3.14
FillStyle =0
FillColor =vbBlue
Circle)Me.ScaleWidth /2, Me.ScaleHeight /2(, 1000, ,) -PI(,) -PI /2(
بين القطر Ratio ، استخدم المتغير لوضع النسبة Ellipse ولرسم الق طع المكافئ
العمودي والافقي:
'القطر العمودي يعادل ضعف القطر الافقي
Circle)Me.ScaleWidth /2, Me.ScaleHeight /2(, 1000, , , , 2
VB5فهي اقوى طرق الرسم والتي ظهرت منذ الاصدار ، PaintPicture اما الطريقة
وتطلب منك Picture الغرض من هذه الطريقة هو رسم صور تابعة للكائن او الخاصية
متغيرات ! لا تخف وتجنب استخدامها لكثرة المتغيرات، فالمطلوبة هي الثلاث 10
الاولى اما الباقية فهي اختيارية، بالنسبة للمتغيرات فالأول هو أسناد الصورة، والاربع

التالية تحدد بها المنطقة التي سترسم الصورة عليها، والاربعة التالية تحدد
المنطقة التي تريد رسمها فعلا من الصورة الاصلية، والمتغير الاخير يحدد اسلوب
رسم الصورة على الهدف، وهو يتطابق تماماً مع ما اوضحته سابقاً حول خاصية
.DrawMode
من فعل اشياء آثيرة على الصور، آ قلبها، عكس PaintPicture تمكنك الطريقة
مثال يعرض لك Codes.ZIP الوانها، تمديدها، تحريكها .... الخ، تجد في ملف الكتاب
وهذا الجزء الاساسي منه: PaintPicture تطبيقات عملية على الطريقة

```
Private Sub Form_Paint()
Cls
PaintPicture Picture1.Picture, 0, 0, lWidth1, _
lHeight1, lX2, lY2, lWidth2, lHeight2, iDrawMode
End Sub
```

موجود Printومازال

لا يزال Print القديمة، فالامر BASIC محتفظا بسمات لغة Visual Basic مازال
. ليس هذا فقط، بل مازالت الصيغ VB6 موجود منذ منتصف الستينات حتى الاصدار
Visual Basic" والعادية مدعومة في :;القديمة آالفواصل المنقوطة "
Print"فاصلة" ,"عادية"
Print"فاصلة" ;"منقوطة"
تقني ا- لا يعتبر–مصنف ضمن طرق الكائنات، الا انه Print **ملاحظة:** رغم ان
حتى تتزامن Visual Basic طريقة. فهو حالة خاصة تعمدها مطوروا
BASICالتوافيقية مع لغة .
، هو نفس القيم الموجودة في الخاصية Print نوع وحجم الخط الناتج من الامر
تفيدان لمعرفة ارتفاع وعرض النص وتختلف TextWidth و TextHeight . الدوال Font
باختلاف نوع وحجم الخط لتتمكن من اختيار الاحداثي المناسب لبدء الكتابة آما
في الكود التالي الذي يكتب النص في وسط النافذة:

```
Dim sText As String
Font.Size =20
sText"ترآي العامري" =
CurrentX ) =ScaleWidth -TextWidth)sText / ((2
CurrentY ) =ScaleHeight -TextHeight)sText / ((2
```



```
Print sText
```

اخيرا، جميع المخرجات النصية عبارة عن نقاط تتشكل في صورة حالها آحال طرق
هي المسئولة عن حفظ المعلومات ا لكاملة لهذهImage الرسم، والخاصية
المخرجات.

احداث النموذج

يحتوي على احداث، معظم Visual Basic نافذة النموذج هي اآثر أسناد من أسنادات
احداثها تم شرحها في فقرة "الاحداث المشترآة ". اما الاحداث الخاصة بها فتفجر
من بداية تحميل النافذة حتى اغلاقها بهذا الترتيب:
<) -Deactivate (< -Paint < -Activate < -Resize < -Load < -Initialize
.Terminate < -Unload < -QueryUnload
الا Paint فلا يتم تفجيره بعد الحدث Deactivate **ملاحظة:** بالنسبة للحدث
في حالة قيام المستخدم بتحديد نافذة اخرى في نفس البرنامج،
واذا عاد المستخدم الى النافذة الاولى، فان السلسة السابقة تبدأ
-Activate < -Paint <من الحدث ...

الحدثInitialize :

يتم تفجير هذا الحدث بمجرد استخدام أسناد النموذج في آاوادك او انشاء نسخة
بتفجير هذا الحدث مبكرا جدا أي قبل Visual Basic جديدة من أسناد النموذج، يقوم
انشاء نافذة النموذج ووضع الادوات عليها:

```
Form2 'التابع لنموذج Initialize يتم تفجير الحدث
Dim X As Form2
Set X =New Form2
```

قد تستفيد من هذا الحدث لتعيين قيم ابتدائية للمتغيرات التابعة لنافذة النموذج
قبل انشاء النافذة:

```
Dim sUserName As String
Private Sub Form_Initialize()
= "مستخدم جديد"sUserName
49
End Sub
```

الحدثLoad :

بمجرد البدء في عملية تحميل النافذة باستخدام الدالة Load يتم تفجير الحدث
:Load

```
Load Form2
```

او حتى عند قيامك باستخدام احد خصائصها او استدعاء طرقها:

```
Form2 'التابع لنموذج Load يتم تفجير الحدث
Caption 'قبل تعديل قيمة الخاصية
Form2.Caption = "النافذة الثانية"
```

لا يتسبب في ظهور النافذة فهو يقع عند Load من الضروري معرفة ان الحدث
تحميل وانشاء النافذة فقط، فلا تحاول استخدام الاوامر التابعة للواجهة آ
او طرق الرسم .... الخ. قد تستفيد من هذا الحدث بوضع قيم ابتدائية SetFocus
آ:

```
Private Sub Form_Load()
Text1.Text =sUserName
End Sub
```

الحدثResize :

يتم تفجيره او آلما قام Resize بمجرد ان تظهر نافذة النموذج، فان الحدث
المستخديم بتحجيم النافذة وتغيير حجمها، قد تستخدم هذا الحدث بكثرة عند
رغبتك في محاذاة الادوات او تغيير حجمها آلما قام المتسخدم بتغيير حجم النافذة:

```
Private Sub Form_Resize()
' توسيط الاداة على النافذة
Command1.Move )Me.ScaleWidth -Command1.Width / (2, _
```

)Me.ScaleHeight -Command1.Height / (2
End Sub
50
Activateالحدث :
- او بمجرد آون النافذة Resize بعد الحدث–يتم تفجير الحدث بمجرد ظهور النافذة
. مع ذلك، لن يتم تفجير الحدث اذا انتقل Active Window هي النافذة النشطة
المستخدم من برنامج آخر الى برنامجك، أي أن هذا الحدث لا يتم تفجيره إلا عند
–التنقل بين نوافذ برنامجك فقط . قد يفيدك هذا الحدث في تغيير محتويات النافذة
آتحديث البيان ات- بمجرد قيام المستخدم بتغيير محتويات نافذة اخرى في نفس
البرنامج:
Private Sub Form_Activate()
Label1.Caption =Form2.Text1.Text
End Sub
Paintالحدث :
يتم تفجير هذا الحدث آلما دعت الحاجة الى اعادة رسم النافذة، فلو قمت بوضع
له Paint النافذة س فوق النافذة ص ومن ث م تعود الى النافذة س، فان الحدث
نصيب من الوقوع، آذلك عندما تخفي اجزاء من النافذة ومن ثم تظهرها سيتم
تفجير الحدث . من الضروري جدا جدا اخبارك بانه في حالة آون قيمة الخاصية
لن يتم تفجيره حتى تحج البقرة Paint فان الحدث True تساوي AutoRedraw
على قرونه !! افضل آواد يمكنك وضعها بين سطور هذا الحدث هي آواد الرسم،
الكود التالي يرسم دائرة تغطي معظم اجزاء النافذة:
Private Sub Form_Paint()
Cls
FillStyle =0
Circle)ScaleWidth /2, ScaleHeight /2(, _
IIf)ScaleWidth < ScaleHeight, ScaleWidth, ScaleHeight / (2, 0
End Sub
في Paint من المفيد ان اذآر هنا بان تغيير حجم النافذة يؤدي الى تفجير الحدث
لا Paint حالة ان قام المستخدم بتكبير الحجم، اما عند تصغير الحجم فان الحدث
يتم تفجيره، وذلك لانه لا توجد حاجة لاعادة رسم اجزاء من النافذة، فقد تلاحظ في
الكود السابق انك اذا قمت بتصغيير حجم النافذة، فان الدائرة لن يتم اعادة رسمها،
لاعادة رسم الدائرة هي طريق الحدث Visual Basic والفكرة الذآية التي قد تجبر
:Resize
51
Private Sub Form_Resize()
Form_Paint
End Sub
رغم ان الكود السابق صحيح، الا انه لا يخرج من اصابع م برمج حريف، والسبب ان
سيتم تنفيذه مرتين آلما قام المستخدم بتكبيير حجم النافذة، Paint الحدث
فالأولى بسبب اعادة الرسم والثانية بسبب الاستدعاء الموجود في الحدث
مباشرة بل Fomr_Paint ، لذلك تجد ان المبرمج الذآي لا يستدعي الحدث Resize
Refreshليفعله عند وقت الحاجة باستخدام الطريقة : Visual Basic يترك الامر ل
Private Sub Form_Resize()
Me.Refresh
End Sub
تكون نافذة النموذج جاهزة لاستقبال الاحداث Paint بعد الحدث التلقائي الاخير
وغيرها، اما في حالة عدم وجود Click الخاصة لباقي الادوات او احداثها الاخرى آ
الخاص بنافذة النموذج سيتم GotFocus أي اداة قابلة لاستقبال الترآيز، فان الحدث
تفجيره فورا.

Deactivateالحدث :
ويتم تفجيره بمجرد ان ينتقل الترآيز الى نافذة اخرى Activate هو عكس الحدث
تابعة لبرنامجك فقط . قد ينفذ هذا الحدث ايضا في حالة الاخفاء المؤقت للنافذة
Falseالى . Visible او تعديل قيمة الخاصية Hide باستخدام الطريقة
QueryUnloadالحدث :
عندما تكون النافذة على وشك الازالة النهائية من QueryUnload يتم تنفيذ الحدث
وليس الاخفاء المؤقت . يمكنك هذا الحدث من الاستعلام عن الطريقة التي–الذاآرة
. المزيد ايضا، UnloadMode تسببت في اغلاق النافذة عن طريق المتغير المرسل
الى المتغير True تستطيع الغاء فكرة اغلاق النافذة عن طريق اسناد القيمة
، فالكود التالي لن يمكن المستخدم من اغلاق النافذة باستخدام Cancel المرسل
" الموجود في اعلى النافذة:Xاو الزر اغلاق " Control Box صندوق التحكم

```
Private Sub Form_QueryUnload)Cancel As Integer, UnloadMode As Integer(
If UnloadMode =vbFormControlMenu Then
Cancel =True
```



```
End If
End Sub
```

موجودة في تعليمات UnloadMode طرق الاستعلام الاخرى عن قيم المتغير
.MSDN
Unloadالحدث :
هو Unload ان لم تقم بالغاء عمل ية اغلاق النافذة في الحدث السابق، فان الحدث
الحدث التالي، معطيك فرصة اخيرة لالغاء عملية اغلاق النافذة عن طريق نفس
فهو غير موجود. UnloadMode ، اما بالنسبة للمتغير Cancel المتغير المرسل
Terminateالحدث :
يتم تفجير هذا الحدث بمجرد موت أسناد النموذج، م وضوع موت الكائنات هو احد
OOPفقرات الفصل الخامس "البرمجة أسنادية التوجه ."

Menusالقوائم

لنوافذ النماذج وقت التصميم عن Menu من تصميم قائمة Visual Basic يمكنك
، حدد نافذة النموذج ثم اختر الامر Menu Editor طريق صندوق الحوار محرر النماذج
. واذا آنت تعاني من آثرة اعادة تكرار تعبئة Tools من قائمة...Menu Editor
عن طريق Template Menu محتويات القوائم، تستطيع استخدام قوالب القوائب
VB6 Template Managerمدير القوالب . Add-In الاضافة
التي تمثل Caption مبدئيا، آل وحدة من وحدات القائمة تحتوي على الخاصية
" لوضع خط تحت الحرف الذي يليه&النص الظاهر على القائم ة، استخدم الرمز "
Alt حتى تمكن المستخدم من الوصول الى الامر في القائمة بالضغط على المفتاح
الرمز "-" فقط، فان Caption والحرف الذي يلي الرمز، واذا أنت قيمة الخاصية
تمثل الاسم الب رمجي Name القائمة ستكون عبارة عن خط فاصل . اما الخاصية
للقائمة والذي تنطبق عليه نفس شروط الادوات في التسيمة، فالقائمة ماهي الا
أداة لكن من نوع خاص، فبامكانك كتابة آاواد تعدل في خصائص القائمة وقت
التنفيذ:

```
mnuFile.Caption = "ملف&"
mnuEdit.Enabled =False
```


موجودة في القوائم وتؤثر حتى في القوائم Enabled و Visible آما ان الخصائص
تحدد ما اذا آنت تريد وضع علامة اختيار Checked الفرعية التابعة له ا. والخاصية
فهي تمكن القائمة من عرض WindowList بجانب عنوان القائمة . اما الخاصية
MDIجميع النوافذ المحضونة في النافذة من النوع .
Pop-Up Menusالقوائم المنبثقة :

اذا نقرت بزر الفأرة الايمن على أي أسناد، فان قائمة صغيرة ستظهر لك . هذه
عن طريق Visual Basic . تستطيع تطبيقها في Pop-Up Menu القائمة تسمى
مع تحديد القائمة التي تود عرضها: PopupMenu الامر

```
Private Sub Form_MouseDown)Button As Integer, Shift As Integer, _
X As Single, Y As Single(
If Button And vbRightButton Then
PopupMenu mnuView
End If
End Sub
```

آما يمكنك عرض قائمة تابعة لنافذة نموذج اخرى:

```
PopupMenu frmMain.mnuHelp
```

الادوات الداخلية

فيما يلي عرض ملخص لجميع الادوات الداخلية الموجودة في صندوق الادوات
Labelوالبداية مع أداة العنوان : ToolBox

Labelأداة العنوان

حيث تعرض النص Windowless Controls اداة العنوان من الادوات المعدومة النوافذ
" قبل احد&التابعة لها، في حالة كتابة الرمز " Caption الموجود في الخاصية
الحروف في هذه الخاصية، فان خط صغير يتم تسطيره تحت ذلك ا لحرف يمكن
المستخدم من نقل التركيز الى الاداة التي تلي أداة العنوان في الخاصية
] وذلك الحرف، تستطيع الغاء الخدمة Altاذا ضغط على المفتاح [TabIndex
Falseالى . UseMnemonic السابقة بتحويل قيمة الخاصية
54
" على الأداة وأنت قيمة الخاصية&**ملاحظة:** اذا اردت عرض الرمز "
، فيشترط كتابة الرمز مرتين. True تساوي UseMnemonic
تحاذي النص Alignment تظهر حدود حول الاداة، والخاصية BorderStyle الخاصية
، من اليمين الى 0-Left Justify الموجود في الاداة اما من اليسار الى اليمين
فهي WordWrap . اما الخاصية 2-Center او في الوسط 1-Right Justify اليسار
مفيدة جدا للنصوص الطويلة حيث تقوم بازاحة النص الى سطر جديد آلما وصل
تحدد ما اذا آنت تريد جعل أداة العنوان شفافة BackStyle حدود الأداة . الخاصية
بحيث تظهر الادوات التي خلفها او لا.
بالاضافة الى عرض النصوص، يوجد استخدام جميل لاداة العنوان اطبقه بك ثرة في
برامجي، حيث اضع مجموعة ادوات العنوان على النافذة التي تحتوي على صورة
لكل أداة، ولحبك الحيلة أقوم بوضع Click لازرار واقوم بكتابة بعض الآواد في الحدث
لكل أداة مما يوحي للمستخدم ان الازرار الموجودة على الصورة ToolTip تلميح
حقيقية.

TextBoxأداة النص

بشكل Windows من آثر الادوات استخداما في تطبيقات Text Box أداة النص
عام، فهي الوسيلة المثلى للتفاعل مع المستخدم والحصول على قيم المدخلات
منه. بعد ان تضيف أداة نص جديدة على النافذة، امسح النص الابتدائي لها عن
. واذا اردت منع المستخدم من تغيير محتويات أداة النص، Text طريق الخاصية
تحدد MaxLength تفي بالغرض . آما ان الخاصية Locked للخاصية True فالقيمة
العدد الاقصى من الحروف التي يمكن ان يكتبها المستخدم . تستطيع تحديد حرف
معين آالنجمة "*" لتظهر بمقدار عدد الحروف المكتوبة عن طريق الخاصية
، ومن الواضح ان الغرض الرئيس لها لكلمات السر. PasswordChar
، فان المستخدم لن يتمكن PasswordChar **ملاحظة:** اذا استخدمت الخاصية
من القائمة Copy من سرقة النص المكتوب على الأداة باختيار الامر
المنسدلة بعد النقر بزر الفأرة الايمن على أداة النص، لأن ذاآرة

لا تنسى الغاء اوامر النس خ والقص من القائمة Visual Basic
السابقة. اما لو انشأت قوائم بها اوامر نسخ ولصق، فذاآرتك هي
المسؤولة عن الغاء او عدم تمكين هذه الوظائف.
55
لتمكن المستخدم من تحرير النص على عدة سطور، MultiLine استخدم الخاصية
فهي تتحكم بظهور او اخفاء اشرطة التمرير. ScorllBars ولا تنسى الخاصية
وقيمة الخاصية True هي MultiLine **ملاحظة:** اذا أتت قيمة الخاصية
، فان النص الذي يكتبه 2-Vertical او 0-None هي ScorllBars
المستخدم سيتم ازاحته الى سطر جديد بمجرد الوصول الى حدود
لاداة العنوان. WordWrap آالخاصية–الاداة
التابعة لاداة النص هي خصائص Run Time Properties من خصائص وقت التنفيذ
وطول SelStart تمكنك من تحديد نص معين، حيث تضع نقطة البداية في الخاصية
. الكود التالي يقوم بتحديد النص بمجرد انتقال SelLength التحديد في الخاصية
الترآيز الى أداة النص:

```
Private Sub Text1_GotFocus()
Text1.SelStart  =0
Text1.SelLength  =Len)Text1.Text(
End Sub
```

. اما الخاصية SelText واذا اردت معرفة او استبدال النص المحدد فاستخدم الخاصية
فهي تمثل آامل النص الموجود في الاداة سواء أن محددا او لا، فلو اردت Text
اضافة نص الى الاداة دون حذف النص الموجود بها فاأكتب شيئا مثل:
Text1.SelText = "نص اضافي"
من الضروري التنويه هنا بان المستخدم لن يستطيع استخدام مفتاح الجدولة
[TAB] اثناء الكتابة في خانة النص، والسبب في ذلك منطقي، فالمفتاح [TAB]
يؤدي الى انتقال الترآيز الى الادوات الاخرى، تستطيع اللتفاف حول هذه المشكلة
لجميع الادوات ومن ثم اعادتها: TabStop البسيطة بالغاء الخاصية

```
Private Sub Text1_GotFocus()
On Error Resume Next
Dim ctrl As Control
For Each ctrl In Controls
ctrl.TabStop  =False
```

56

```
Next
Err.Clear
End Sub
Private Sub Text1_LostFocus()
On Error Resume Next
Dim ctrl As Control
For Each ctrl In Controls
ctrl.TabStop  =True
Next
Err.Clear
End Sub
```

العرب هي عدم ظهور Windows 2000, XP من المشاآل التي تواجه مستخدمي
احيانا- عند نسخها من أداة النص والصاقها الى–الحروف العربية بالشكل المطلوب
برنامج آخر، والسبب في ذلك يتعلق بتوزيع صفحات المحارف التابعة لترميز
، لا اريد ASCIIما زال مبني على جدول Visual Basic لان ترميز أدوات UNICODE
ان افصل في الموضوع آاثر من ذلك حتى لا نخرج عن مجال الفقرة، ولكنك
تستطيع حل هذه المشكلة بتغيي ر اللغة الى اللغة العربية بالضغط على الازرار
[ ] او عمل ذلك برمجيا قبل عملية النسخ او القص:Alt+SHIFT

```
Declare Function LoadKeyboardLayout Lib "user32 "Alias _
"LoadKeyboardLayoutA) "ByVal pwszKLID As String, ByVal _
flags As Long (As Long
Sub BeforeCopyOrCut()
LoadKeyboardLayout "00000401", 1
End Sub
```

السيطرة على المدخلات:

المشكلة التي اود ان اوضحها هو اننا حين نبرمج نتوقع ادخالات معينة من المستخدم. فمثلا، وضعت أداة نص لتجعل المستخدم يكتب عمره فبكل تأكيد ستتوقع ان يكون العمر قيمة عددية، لكن ماذا لو ادخل المستخدم حروفا؟ فا نه من المؤآد أن منطق سير وسلوك تنفيذ البرنامج سيتأثر في افضل الاحوال هذا اذا لم . لذلك ستضطر لكتابة آآواد اضافية لتضمن ان Run Time Error تظهر رسالة الخطأ

57

أداة النص لا تحتوي الا على اعداد، ولعل الحدث المناسب لكتابة آود التحقق هو KeyPressحدث :

```
Private Sub Text1_KeyPress )KeyAscii As Integer(
If KeyAscii < 48 Or KeyAscii > 57 Then
'المفتاح المدخل ليس عدد
KeyAscii =0
End If
End Sub
```

مهلا مهلا اخي الكريم، عالم البرمجة لعبة عقلية ومنطقية، والامور فيه لا تتم بالسهولة التي تتو قعها! لانه ما زالت هنالك امكانية ان يدخل المستخدم حروف في أداة النص وهي باختصار : عن طريق لصق قيم حرفية من الحافظة أي بالضغط . فورا ستكون اجابتك الذآية جدا هي ان نكتب آآواد +V Ctrl على المفتاحين نمنع فيه المستخدم من اجراء عملية اللصق . صح KeyDown اضافية في حدث لسانك! لكنك نسيت طريقة اخرى للصق وهي عن طريق القائمة المختصرة التي الى أداة النص والتي تظهر عن طريق زر الفأرة الايمن، والتي Visual Basic يضيفها من خلالها يستطيع المستخدم ان يلصق النصوص!

لا توجد مشكلة الا ولها حل فهذا عالم البرمجة مشاآل وحلول . من وجه ة نظ ري الشخصية، ارى ان افضل مكان -اقصد حدث- للتحقق من نوع قيمة النص المدخل ، لكن المشكلة فيه انه يتطلب تصريح متغير ين عامين للعودة Change هو الحدث بالقيمة القديمة لأداة النص اذا أتت القيمة الجديدة ليست عددية:

```
Dim OldText As String
Dim OldSelStart As Long
Private Sub Text1_GotFocus()
'عندما يكون الترآيز على الاداة
'لابد من حفظ قيمتها
OldText =Text1.Text
OldSelStart =Text1.SelStart
End Sub
Private Sub Text1_Change()
If Not IsNumeric)Text1.Text (Then
'المفتاح المدخل ليس رقم
```

58

```
'قم باعادة عرض القيمة القديمة
Text1.Text =OldText 'توجد مشكلة خطيرة هنا
Text1.SelStart =OldSelStart
Else
'القيمة المدخلة رقمية اذا
```

```
`قم بحفظها
OldText =Text1.Text
OldSelStart =Text1.SelStart
End If
End Sub
```

آما تلاحظ، في الكود السابق لك مني ضمان ان المستخدم لن يستطيع ادخال الا الارقام لكن في احد السطور أكتبت التعليق "توجد مشكلة خطيرة هن ا" والسبب انه بتنفيذ Visual Basic ، سيقوم OldText= Text1.Text عندما يتم تنفيذ السطر من جديد ! أي ان هذا الاجراء سيتم تنفيذه آما يعرف في Text1_Change الاجراء وهو احد اساليب الخوارزميات التراجعية Recursivly عالم البرمجة تراجعي . وحتى تتفادى هذه المشكلة استخدم متغير ستاتيكي يمنع حدوث Recursion ذلك:

```
Private Sub Text1_Change()
`متغير يمنع استدعاء الاجراء تراجعيا
Static bExitNow As Boolean
If bExitNow Then Exit Sub
If Not IsNumeric)Text1.Text (Then
`المفتاح المدخل ليس رقم
`قم باعادة عرض القيمة القديمة
bExitNow = True
Text1.Text =OldText `ذهبت المشكلة الخطيرة التي أتت هنا
bExitNow =False
Text1.SelStart =OldSelStart
Else
`القيمة المدخلة رقمية اذا
`قم بحفظها
OldText =Text1.Text
OldSelStart =Text1.SelStart
```

59

```
End If
End Sub
```

الخاص Caret مازالت توجد مشكلة اخرى وخطيرة ايض ا! وهي تتعلق بموقع المؤشر بأداة النص . فالكود السابق لا يق وم بحفظ موقع المؤشر الا في حالة تغيير الق يمة لأداة النص مما يتسبب في مشاآل لا نهاية لها عندما يقوم المستخدم بتغيير مكان المؤشر دون تغيير القيمة آتحريكه بلاسهم في لوحة المفاتيح او بزر الفأرة .
والحل عن طريق حفظ قيمة موقع المؤشر في حالة حدوث ذلك:

```
Private Sub Text1_KeyUp )KeyCode As Integer, Shift As Integer(
OldSelStart =Text1.SelStart
End Sub
Private Sub Text1_MouseUp )Button As Integer, Shift As Integer, _
X As Single, Y As Single(
OldSelStart =Text1.SelStart
End Sub
Private Sub Text1_KeyDown )KeyCode As Integer, Shift As Integer(
OldSelStart =Text1.SelStart
End Sub
Private Sub Text1_Click()
OldSelStart =Text1.SelStart


End Sub
```

بكل تأكيد لو لم يكن هناك حل آخر لما آنت قد عر ضت المشكلة من البداية، لان
الحل يتم باختبار القيمة بعد ان ينتهي المستخدم من ادخال القيمة وليس في اثناء
Visual Basic الذي ظهر آحل لمبرمجي Validate ذلك. ويتم ذلك عن طريق الحدث
يعمل Validate لمواجهة مثل هذه المشاآل بالتحديد مع اصداره السادس . حدث
، لمعرفة آيف يتم ذلك، راقب هذا CausesValidation بالتكامل مع الخاصية
Visual Basic ، يقومX الى الاداة Text1 السيناريو: عندما ينتقل التركيز من أدا النص
True، واذا أتت قيمتها ،X التابعة للاداة CausesValidation باختبار قيمة الخاصية
60
معطيا المبرمج Text1 التابع لأدا النص Validate بتنفيذ الحدث Visual Basic يقوم
فرصة لاختبار نوع القيمة. فتستطيع اختصار هذه المقالة بهذا الكود:

```
Private Sub Text1_Validate)Cancel As Boolean(
If Not IsNumeric)Text1.Text (Then
`المفتاح المدخل ليس رقم
Cancel  =True
End If
End Sub
```

حتى نلغي عملية الادخال True الى Cancel فتلاحظ ا ننا قمنا بتغيير قيمة المتغير
. وقد يسأل سائل ويقول لماذا لا Text1 من المستخدم ونعود بالترآيز الى أدا النص
LostFocus ؟ والجواب هو ان الحدث Validate بدلا من LostFocus نستخدم الحدث
غير مرن ! أي انك تحتاج الى آاواد اضافية لتع ديل الخيارات (فلا يوجد به المتغير
سيقوم دائما باختبار القيمة رغم انك لا تود LostFocus) المزيد ايضا، الحدث Cancel
اختبار القيمة دائم ا، مثلا عندما يقوم المستخدم بالضغط على الزر الغاء الامر
الخاص بصندوق الحوار الذي صممته. Cancel

CommandButtonزر الاوامر

. لا يوجد CommandButton بعد اداتي العنوان والنص تبدأ باستخدام زر الاوامر
، الاولى تضع حد اضافي Cancel و Default الكثير لاخبرك به عنها الا الخاصيتان
حول الزر تمكن المستخدم من تنفيذ زر الاوامر بمجرد الضغط على المفتاح
Picture]. وبالنسبة لخصائص الصور ، [ESC] والثانية مرافقة للمفتاح [ENTER
فلن تتمكن من رؤية نتائجها حتى تغير قيمة DownPicture و DisablePicture
. 1-Graphical الى Style الخاصية
" دائما آما تتأثر أداة&فهي تتأثر بالرمز " Caption **ملاحظة:** بالنسبة للخاصية
UseMnemonic بهذا الرمز عندما تكون قيمة خاصيتها Label العنوان
Trueتساوي .

CheckBoxأداة الاختيار

تعطي هذه الاداة فرصة للمستخدم لتحديد اختيار معين اما بتفعيله او لا، قيمة
او 0-UnChecked، 1-Checked والتي تكون اما Value التفعيل تحتجز في الخاصية
61
، بالنسبة للقيمة الاخيرة، فيعتقد معظم المبرمجين انها تشبه القيم ة 2-Grayed
للخاصية False % لان القيمة 100 ، وهذا الاعتقاد خاطئ Enabled للخاصية False
تمكنه من 2-Grayed تمنع المستخدم من التعامل مع الاداة، بينما القيمة Enabled
ذلك، تستطيع استخدام هذه الخاصية في حالات ما بين تحديد الاختيار او لا،
آتحديد مجموعة اختيارات فرعية آلها او بعضها او عدم تحديدها، في املف
تجد مثالا تطبيقيا على هذه القيمة. Codes.ZIP
يتم تفجيره عند قيامك بتعديل Click من الضروري ان تضع في ذهنك بان الحدث
حتى لو لم يقم المستخدم بالنقر على الاداة . وبالنسبة Value قيمة الخاصية
فلن تتمكن من رؤية DownPicture و Picture، DisablePicture لخصائص الصور
. 1-Graphical الى Style نتائجها حتى تغير قيمة الخاصية
" تماماً آما تتأثر أداة&فهي تتأثر بالرمز " Caption **ملاحظة:** بالنسبة للخاصية

بهذا الرمز. CommandButton زر الاوامر

OptionButtonزر الاختيار

وهو شبيه بأداة Radio Button يسمى احيانا بزر الراديو OptionButton زر الاختيار
. آذلك لن False او True تكون اما Value ، الا ان قيمة الخاصية CheckBox الاختيار
لاآثر من زر اختيار واحد في نفس True الى Value تستطيع جعل قيمة الخاصية
المجموعة -أي نفس الاداة او نافذة النموذج الحاضنة، لذلك يفضل وضع هذه الازرار
وترتيبها على مجموعات تناسب تصنيف الاختيارات . Frame في داخل اداة الاطار
فلن تتمكن من DownPicture و Picture، DisablePicture وبالنسبة لخصائص الصور
. 1-Graphical الى Style رؤية نتائجها حتى تغير قيمة الخاصية
" تماماً آما تتأثر أداة&فهي تتأثر ب الرمز " Caption **ملاحظة:** بالنسبة للخاصية
بهذا الرمز. CommandButtonزر الاوامر

ListBoxأداة القائمة

تعرض لك هذه الاداة مجموعة من النصوص في داخل صندوق يحتوي على اشرطة
تقوم بفرز محتويات الاداة فرز تصاعدي بالاستناد Sorted . الخاصية ScrollBars تمرير
للاسفل- اسفل–على حروفها الابجدية، مع العلم ان الحروف العربية ستكون
الحروف الانجليزية . تستطيع عرض آل محتويات الاداة على شكل اعمدة وتلغى
شريط التمرير العمودي لتظهر شريط التمرير الافقي، سيكون عدد الاعمدة هو
62
رغم اني لا احبذ هذه الطريقة لعرض Columns نفس العدد الموجود في الخاصية
محتويات الاداة.
لن يكون دقيقا آما هو الحال مع الادوات الاخرى، Heightتغيير ارتفاع الاداة
يحاول تعديل ارتفاع الاداة حتى تعرض السطر آاملا في Visual Basic والسبب ان
القائمة، فلن تستطيع عرض نصف السطر الا اذا قمت بالغاء المحاذاة التي يفعلها
Falseتساوي . IntegralHeight عن طريق جعل قيمة الخاصية Visual Basic
او وقت List تستطيع تعبئة محتويات الاداة في وقت التصميم عن طريق الخاصية
AddItemالتنفيذ باستخدام الطريق :

```
List1.AddItem "الاول"
List1.AddItem "الثاني"
List1.AddItem "الثالث"
```

اذا آنت ستضيف مئات او الاف العناصر وقت التنفيذ، فينصح باخفاء الاداة مؤقتا وبعد
اضافة العناصر تعيد اظهارهامن جديد، وذلك لأن الاداة تعيد رسم نفسها تلقائياً مع
اضافة أي عنصر إليها، مما يتسبب في بطء التنفيذ وآذلك ارتعاش الاداة:

```
List1.Visible  =False
For X  =0 To 10000
List1.AddItem X
Next
List1.Visible  =True
```

Sorted العناصر الجديدة تضاف الى نهاية سلسلة العناصر -اذا أتت قيمة الخاصية
- مالم تحدد موقعها بنفسك: False تساوي

```
List1.AddItem "الثالث"
List1.AddItem "الاول", 0
List1.AddItem "الثاني", 1
```

تذآر ان تحديد موقع العنصر يؤثر في ترتيب العناصر الاخ رى. بامكانك حذف العنصر
لحذف جميع العناصر: Clear او الطريقة RemoveItem باستخدام الطريقة

```
List1.RemoveItem 0
List1.Clear
```

63
ان لم 1تعود بقيمة العنصر المحدد في الاداة، وتعود بالقيمة - ListIndex الخاصية

Text يكن هناك أي عنصر محدد، بامكانك تعديل قيمة الخاصية ايضا، ام ا الخاصية
فهي تعود بنص العنصر المحدد:

```
List1.ListIndex =0
Print List1.Text
```

فهي تعود بعدد جميع العناصر الموجودة، والتي ListCount بالنسبة للخاصية
التي تمكنك من الوصول الى العنصر: List تستخدم بكثرة مع الخاصية

```
For X =0 To List1.ListCount
Print List1.List )0(
Next
```

فهي تمكن المستخدم من تحديد عدة عناصر MultiSelectبالنسبة للخاصية
باستخدام–او عدة عناصر غير متتالية 1-Simple متتالية في الاداة اذا أتت قيمتها
، وتستطيع معرفة العنصر المحدد عن 2-Extended]- اذا أتت قيمتها Ctrlالمفتاح [
، ه ذا الكود ينقل جميع العناصر المحددة من اداة القائمة Selected طريق الخاصية
الى اخرى:

```
Private Sub cmdMoveLeft_Click()
Dim iCounter As Integer
iCounter =0
Do While iCounter < =lstRight.ListCount -1
If lstRight.Selected)iCounter = (True Then
lstLeft.AddItem lstRight.List)iCounter(
lstRight.RemoveItem iCounter
iCounter =iCounter -1
End If
iCounter =iCounter +1
Loop
End Sub
```

64

CheckBoxes ايضا لمعرفة ادوات الاختيار Selected تستخدم نفس الخاصية
تساوي Style المحددة بجانب اسماء العناصر والتي تظهر اذا أتت قيمة الخاصية
.1-CheckBox
يتم تفجيره بمجرد قيام المستخدم باستخدام اشرطة التمرير Scroll اخيرا، الحدث
التابعة للاداة. ScrollBars

ComboBoxأداة القائمة

السابقة موجودة في الاداة ListBox معظم الخصائص والطرق التابعة للاداة
قياسية وتحتوي ListBox عبارة عن أداة ComboBox ، وذلك لان الاداة ComboBox
اضافية في اعلاها، تستطيع عرض خانة النص بعدة طرق TextBox على خانة نص
0-Dropdown تساوي Style . اذا أتت قيمة الخاصية Style باستخدام الخاصية
فان اداة النص ستظهر مع س هم يؤدي النقر عليه الى ظهور الجزء الثاني Combo
1-Simple ، واذا أتت قيمة الخاصية تساوي ListBox الشبيه بالاداة–من الاداة
2فكلا الجزئين سيظهران للمستخدم، اما اذا أتت قيمة الخاصية تساوي - Combo
فهي مثل القيمة الاولى باستثناء ان المستخدم لن يتمكن من Dropdown List
الكتابة في خانة النص.
Scorll اما الحديث عن الاحداث فهي تحتوي معظم الاحداث القياسية والحدث
ان صح التعبير –او Visual Basic ، الا ان فريق التطوير ل ListBox الموجود في الاداة
MouseDownقد نسي اضافة الاحداث ، ComboBox المطور الذي قام ببرمجة الاداة
! اذا صادفته يوما من الايام فارجو ان تنصحه بتناول افطاره MouseUp و MouseMove
Microsoftقبل الذهاب الى مكتبه في .

PictureBoxأداة الصورة

بدون شريط Form على انها نافذة نموذج PictureBox يمكنك اعتبار اداة الصورة

عنوان، فمعظم خصائص، طرق واحداث نافذة النموذج موجودة في الاداة
لذلك لا يوجد داعي لاعادة ذآرها في هذه الفقرة باستثناء الخاصية PictureBox
التي تعدل حجم الاداة بحيث تعرض جميع محتويات الصورة الموجودة في AutoSize
اداة PictureBox . آما ان الاداة Picture الاداة -أي الصورة الموجودة في الخاصية
حاضنة يمكنها ان تحتضن ادوات اخرى في داخلها.

Imageأداة الصورة

، فهي PictureBox هي نسخة مبسطة من الاداة السابقة Image اداة الصورة
الا انها لا تدعم الخصائص المتقدمة للصور Picture تعرض الصور عن طريق خاصيتها
65
Line.... الخ آما انها لا تدعم طرق الرسم ، HasDC، AutoRedraw، DrawMode آ
.... الخ، اذا آنت لا تنوي استخدامها فقد Paint، Resize .... الخ، والاحداث Cirlce
Windowless هي اداة من النوع معدومة النوافذ Image تخسر الكثير! لان الاداة
، آما ان استهلاآها لمصادر PictureBox أي انها اسرع بكثير من الاداة Control
. فاذا أن استخدامك PictureBox اقل بكثير من الاداة System Resources النظام
عوضا عن الاداة Image لادوات الصورة قاصرا على عرض الصور فقط، استخدم الاداة
.PictureBox

ScrollBarsاشرطة التمرير

من محاآاة اشرطة التمرير المنتشرة في VScorllBar و HScorllBar تمكنك الاداتين
لتحديد مجال القيم التي Max و Min . ابدأ بالخاصيتين Windows نوافذ وتطبيقات
، والتي تمثل الموقع الحالي Value يمكنك قرائتها او آتابتها عن طريق الخاصية
للمستطيل المتحرك في شريط التمرير . بعد ذلك، حدد قيمة التغيير البسيط عن
وهي مقدار التغيير في القيمة عندما يقوم SmallChange طريق الخاصية
فهي LargeChange المستخدم بالنقر ع لى احد ازرار اشرطة التمرير، اما الخاصية
مقدار التغيير في القيمة عندما يقوم المستخدم بالنقر على شريط التمرير نفسه.
توجد خاصية تعرض اشرطة الادوات على نوافذ النماذج مباشرة، اما Delphi في لغة
فللاسف لا توجد، اذا آنت تريد وضع اشر طة ادوات عليها، فانت Visual Basic مع
المسؤول الاول والاخير عن ظهورها، اخفائها وتحريك الادوات، وبالنسبة لتحريك
حتى تسهل عملية التحريك PictureBox الادوات فيفضل احتضان الادوات في اداة
عليك:

```
Private Sub Form_Resize()
If Me.ScaleHeight < picMain.Height Then
VScroll1.Move 0, 0, VScroll1.Width, Me.ScaleHeight  -HScroll1.Height
picMain.Left  =VScroll1.Width
VScroll1.Min  =0
VScroll1.Max )- =picMain.Height  -Me.ScaleHeight(
VScroll1.SmallChange  =50
VScroll1.LargeChange  =500
Else
picMain.Left  =0
VScroll1.Move Me.ScaleWidth  +VScroll1.Width
End If
```
66
```
If Me.ScaleWidth < picMain.Width Then
HScroll1.Move VScroll1.Width, Me.ScaleHeight  -HScroll1.Height, _
Me.ScaleWidth  -VScroll1.Width
HScroll1.Min  =0
HScroll1.Max )- =picMain.Width  -Me.ScaleWidth(
HScroll1.SmallChange  =50
HScroll1.LargeChange  =500
```

```
Else
HScroll1.Move 0, Me.ScaleHeight +HScroll1.Height
End If
End Sub
```

ادوات الملفات

غرضها FileListBox و DriverListBox، DirListBox من الواضح ان ادوات الملفات
الاساسي هو عرض ملفات الجهاز، يعيبها انها ادوات قديمة جدا، وشكلها الخارجي
، ورغم انني لا احبذ Windows لا يلائم ادوات عرض الملفات الشائعة لبيئة
على الاقل - تعتبر م ن الادوات الداخلية،–استخدامها، الا انني ملزم بشرحها فهي
وآما يقولون: "يمكنها تفك ازمة!".
DriverListBox في اغلب النوافذ، تعمل هذه الادوات جنبا الى جنب فتبدأ بالاداة
Driveعن طريق الخاصية : DiskDrive وتحدد حرف محرك الاقراص الابتدائي

```
Private Sub Form_Load()
Drive1.Drive " =C:\"
End Sub
```

حتى تغير مسار الادلة الموجودة في الاداة Change ثم تنتقل الى الحدث
في آل مرة يقوم بها المستخدم بتغيير المحرك في الاداة DirListBox
:DriveListBox

```
Private Sub Drive1_Change()
Dir1.Path =Drive1.Drive
End Sub
```

67

من المهم جدا توقع الخطأ في الكود السابق، فلو قام المستخدم باختيار محرك
: مثلا- فرسالة الخطأ ستنفجر في وجهA اقراص لا يوجد به قرص -آالمحرك
المستخدم وتنهي البرنامج:

```
Private Sub Drive1_Change()
On Error Resume Next
Dir1.Path =Drive1.Drive
If Err Then
Drive1.Drive =Dir1.Path
Err.Clear
End If
End Sub
```

بمجرد تغيير مسار الاداة FileListBox والخطوة الاخيرة تغيير محتويات الاداة
ونكون بذلك قد ربطنا الادوات الثلاث: DirListBox

```
Private Sub Dir1_Change()
File1.Path =Dir1.Path
End Sub
```

التابعة للاداة Path **ملاحظة:** يمكنك كتابة المسار مباشرة للخاصية
FileListBoxحالها آحال الاداة : DirListBox

```
Dir1.Path " =C:\Windows"
File1.Path " =C:\Winnt"
```

التي Pattern فهي تحتوي على خاصية مرنة تدعى FileListBox بالنسبة للاداة
تمكنك من تصفية الملفات وتحديد نوع معين من الامتدادات التي تظهر على الاداة:

```
File1.Pattern =".*EXE"
File1.Pattern =".*TXT;.*DOC;.*RTF"
File1.Pattern ="ABC*.*"
```

المزيدArchive ،Normal من عمليات التصفية تمكنك منها الخصائص المنطقية ،
واظهار ReadOnly . فلاخفاء ملفات القراءة فقط System و Hidden، ReadOnly
اآتب الكود البسيط التالي: Hidden الملفات المخفية

File1.ReadOnly =False
File1.Hidden =True
فهي تمثل الملف المحدد في الاداة. FileName اما الخاصية
ComboBox و ListBox **ملاحظة:** ادوات الملفات الثلاثة شبيهه بأدوات القوائم
ايضا. ListIndex و List، ListCount فهي تحتوي على الخصائص

ادوات اخرى

التي Frame من الادوات الاخرى التي أود ان اختم بها هذا الفصل هي أداة الاطار
تستخدم بكثرة لحضن الادوات وتقسيمها إلى مجموعات، ويمكنك التحكم في
. واداة رسم BorderStyle ظهور او اخفاء الحد الخارجي لها عن طريق الخاصية
التي لا يوجد داعي لذآر الشكل الذي ترسمه ولكن ما دعاني لذآره Line الخطوط
Left، Topفهي ليست ، Y2 و X1، X2، Y1 هو خصائص الموقع والحجم التابعة لها
، ولكنها احداثيات نقطتي البداية والنهاية للخط الذي تريد رسمة Height و Width
اشكال مختلفة 6 فتستخدم لرسم شكل من بين Shape اما اداة رسم الاشكال
تستطيع تحديدها عن طريق الخاصية التي تحمل نفس اسم الاداة . وأداة المؤقت
آل فترة Timer التي تمكنك من تنفيذ الاوامر الموجودة في حدثها الوحيد Timer
ثانية. اما أداة 0.001 ووحدتها Interval معينة، تحديد هذه الفترة تضعه في الخاصية
لوضع مستندات OLE فهي تمكنك من استخدام تقنية OLE ربط الكائنات وتضمينها
مختلفة من تطبيقات مختلفة في نافذة النموذج . التحدث عن تفاصيل الاداة او تقنية
خارج نطاق هذا الكتاب. OLE
للأداة Timer **ملاحظة:** لا تحاول الاآثار من الاآواد الطويلة بداخل الحدث
من مصادر Processing فذلك يسبب استهلاك آبير للمعالجة Timer
مما يؤدي الى ابطاء تنفيذ برامجك System Resources النظام
والبرامج الاخرى في نظام التشغيل.

الفصل الثالث

لغة البرمجةBASIC

Visual Basic شخصية اصيلة معتزة بامجادها وتاريخها، فما زالت Visual Basic ان
High التي تصنف من لغات البرمجة العليا BASIC محتفظة بسمات لغة البرمجة
Visual Basicهي روح لغة البرمجة ، BASIC ، لغةlevel programming language
الى جانب مصمم النماذج Visual Basic وهي اللغة التي زادت من اسهم وشعبية
التي ظهرت بها اللغة منذ بداية الستينات Syntax . فمعظم الصيغ Form Designer
. ليس هذا فقط، بل Visual Basic مازالت مدعومة بشكل جيد في احدث اصدارات
Visual Basic اضيفت اليها العشرات من الدوال والصيغ البرمجية حتى تلائم قوة
في امكانياتها. Windows وتحاآي تطبيقات

المتغيرات والثوابت

المتغيرات والثوابت هي اساس أي لغة برمجة . إن استيعاب انواع المتغيرات من
المسائل الضرورية التي تمكنك من اختيار الانواع المناسبة للمتغيرات سواء
لارسالها الى الدوال او لإجراء العمليات الحسابية عليه ا. بودي التحدث عن مبدئا
قابلية الرؤية وعمر الحياة قبل الخوض في تفاصيل المتغيرات.

قابلية الرؤية وعمر الحياة

قابلية الرؤية وعمر الحياة من احد المبادئ الضرورية في جمي ع لغات البرمجة، و
يعتبر لغة برمجة حقيقة تدعم هذان المبدئان. Visual Basic
- للمتغير تمثل قدرة البرنامج على الوصول Scope او المدى– Visibility قابلية الرؤية
الموجود في الكود التالي لا يمكن الوصول اليه X الى المتغير واستخدامه، فالمتغير
MySub1خارج الاجراء :

```
Sub MySub1 ()
Dim X As Integer
X  =20
71
End Sub
Sub MySub2 ()
Print X 'السابق  X لا يمثل المتغير
End Sub
```

للمتغير، فهي تمثل الفترة التي يظل فيها المتغير محتفظا LifeTime اما عمر الحياة
الموجود في الكود السابق، سينتهي ويزال تلقائيا من الذ اآرة X بقيمته، فالمتغير
Visual Basic . ولكي تفهم الاسلوب الذي يتبعه Sub1 بمجرد الخروج من الاجراء
لتطبيق مبدأ قابلية الرؤية وعمر المتغيرات، عليك معرفة انواع المتغيرات من منظور
الرؤية وعمر الحياة:

المتغيرات المحلية الديناميكية:

هي متغيرات تولد مع Dynamic Local Variables المتغيرات المحلية الديناميكية
السطر الذي تعلن عنها فيه داخل الاجراء وتموت بعد نهاية الاجراء مباشرة ويتم
تحرير المساحة التي حجزتها هذه المتغيرات في الذاآرة، وبالنسبة لقابلية الرؤية
فلن تستطيع الوصول الى هذه المتغيرات الى في نفس الاجراء الذي صرح فيه
لتصريح المتغير مع كتابة اسمه ونوعه: Dim المتغير. تستخدم الكلمة المحجوزة
Dim sName As String

Dim iAge As Integer
موجودة في اعلى منطقة الاعلانات Option Explicit اذا أتت الكلمة المحجوزة
، فعليك اللالتزام بالتصريح آما في BAS العامة لنافذة النموذج او ملف البرمجة
مسطورة فيمكنك Option Explicit الصيغة السابقة، وان لم تكن الكلمة المحجوزة
تعريف المتغير مباشرة دون الالتزام بعملية التصريح باسناد قيمة ابتدائية له:
sName = "ترآي العامري"
iAge =99
صحيح ان الكود السابق يوفر عليك عناء تصريح المتغير لا انه غير محبذ بشكل آبير
لدى المبرمجين الجادين، قد يعرض هذا المثال احد الاسباب:
sCompanyName = "الشرآة التجارية"
Print sConpanyName ` 0 الناتج
72
في الكود السابق لن يكون آما هو متوقع "الشرآة Print الناتج من عملية الطباعة
وليس sConpanyName التجارية"، فالمتغير المستخدم في السطر الثاني هو
في برامجك. Bugs وهذا الخطأ آفيل في نمو الشوائب البرمجية sCompanyName
سبب آخر قد يجعلك تحبذ الالتزام بعملية التصريح وهو ان جميع المتغيرات تكون
هو ابطأ انواع Variant ان لم يتم تصريح نوع غير ذلك، والنوع Variant من النوع
المتغيرات آما سيأتي لاحقا.
إلى الاعلان Option Explicit في مثالنا السابق؛ يؤدي فرض الاعلان عن المتغيرات
عن خطأ و توقف البرنامج . وفي جميع الحالات فإن الخطأ في كتابة اسم المتغير أو
اسناد قيمة إلى متغيرات لم يتم الاعلان عنها مسبقاً سيتتسبب في الإعلان عن
خطأ، وسيتوقف البرنامج ايضاً.
خيار يلزمك بعملية التصريح أيIDE **ملاحظة:** توفر لك بيئة التطوير المتكاملة
في جميع وحدات برامجك Option Explicit بكتابة الكلمة المحجوزة
آنوافذ النماذج، ملفات البرمجة .... الخ. لتفعيل الاختيار، حدد
Editor من خانة التبويب Require Variable Declaration الاختيار
Optionsفي صندوق الحوار .
، والحرفي يكون قيمة حرفية0 اخيرا، القيمة الابتدائية للمتغير العددي المصرح هي
Nothingخالية ""، اما الكائنات فهي لا شئ .

المتغيرات المحلية الستاتيكية:

هي مثل Static Local Variables قابلية الرؤية للمتغيرات المحلية الستاتيكية
قابلية الرؤية للمتغيرات المح لية الديناميكية أي لن تتمكن من الوصول اليها الا من
داخل الاجراء المصرح عنها فيه، وبالنسبة لعمر حياة المتغير الاستاتيكي فهو ييبقى
او حتى يموت BAS محتفظا بقيمته حتى نهاية البرنامج اذا أن في ملف برمجة
عوضا Static الكائن التابع له . لتصريح متغير ستاتيكي استخدم الكلمة المح جوزة
Dimعن :
Static bStaticVariable As Boolean
تستطيع جعل جميع المتغيرات التابعة للاجراء ستاتيكية بوضع نفس الكلمة
المحجوزة عند بداية الاجراء:
73

```
Static Sub Counter ()
`جميع المتغيرات التالية ستاتيكية
Dim iCounter As Integer
Dim iCounter2 As Integer
...
End Sub
```

لا تحاول تطبيق الكود السابق آثيرا، فالمتغيرات الستاتيكية ابطأ من المتغيرات
الديناميكية الى جانب قيامها بحجز مواقع هذه المتغيرات في الذاآرة طوال فترة
عمل البرنامج، فلا تحاول استخدامها الا عند الحاجة آالرغبة في تنفيذ اجراء معين

لمرة واحدة مثلا او الاحتفاظ بقيمة المتغير في عداد:

```
Sub PrintData ()
Static bIsPrinting As Boolean
If bIsPrinting Then
Exit Sub
Else
bIsPrinting =True
End If
...
End Sub
Sub Counter ()
Static iCounter As Integer
iCounter =iCounter +1
End Sub
```

لا تطبق الا على المتغيرات المحلية فلا تحاول Static اخيرا، الكلمة المحجوزة استخدامها على متغيرات عامة او على مستوى الوحدة فهي بطبيعتها ستاتيكية.

المتغيرات على مستوى الوحدة:

اقصد الوحدة البرمجية

.... الخ Class او فئة Form او نافذة نموذج BAS المتمثل ة في ملف برمجة Module من الوحدات المكونة للمشروع . يمكنك تصريح متغير على مستوى الوحدة في منطقة الاعلانات العامة للوحدة أي خارج الاجراءات.

قابلية الرؤية لهذا النوع من المتغيرات يكون عام لجميع آواد الوحدة في حالة اPrivate او : Dim استخدام الكلمة المحجوزة

```
Dim sName As String
Dim iAge As Integer
Sub SetData ()
sName = "ترآي العامري"
iAge =99
End Sub
Sub PrintData ()
Print sName
Print iAge
End Sub
```

اما اذا آنت تريد تعريف متغيرات عامة قابلة للوصول من جميع انحاء المشروع ، تفي بالغرض: Public فالكلمة المحجوزة

```
'في ملف برمجة BAS
Public iNumberOfUsers As Integer
'في نافذة نموذج Form1
Public sCurrentUser As String
'في نافذة النموذج Form2
Private Sub Form_Load()
If iNumberOfUsers < =0 Then
Exit Sub
Else
Me.Caption =Form1.sCurrentUser
End If
End Sub
```

فهي مازالت موجودة لضمان Global **ملاحظة:** بالنسبة للكلمة المحجوزة ، وهي تؤدي نفس Visual Basic التوافقية مع الاصدارات القديمة ل ، ولكنك لن تستطيع استخدامها الا Public غرض الكلمة المحجوزة فقط. BAS في ملفات البرمجة

اما عمر الحياة لهذا النوع من المتغيرات فيكون مرافق لعمر حياة الكائن التابع له
آعمر حياة المتغيرات الستاتيكية، وبالنسبة للمتغيرات العامة–والمصرح فيه
، فستظل محتفظة بقيمتها حتى نهاية تنفيذ BAS المصرحة في ملفات البرمجة
البرنامج.

المتغيرات

نستطيع ان نعرف المتغيرات بمنظورين، بالمنظور بغدادي يعرف المتغير على انه
وهو الاهم - يعرف–مجهول س يحتوي على قيمة معينة، اما بالمنظور البرمجي
المتغير على انه قيمة تحفظ في ذاآرة الجهاز . وتختلف المساحة المحجوزة لحفظ
لا يستهلك سوى بايت Byte هذه القيمة باختلاف نوع المتغير، فمتغيرمن النوع
قد يحجز مساحة تصل String واحد من ذاآرة الحاسب، في حين أن متغير من نوع
جيجابايت. 2 الى
Visual Basicوفيما يلي عرض لجميع انواع المتغيرات المدعومة من قبل :

Byteالمتغيرات من النوع :

] وهو 255 ،0يستطيع هذا النوع الاحتفاظ باي قيمة صحيحة ضمن المجال العددي [
أتت Byte بايت . بداية المتغيرات من نوع 1 اصغر انواع المتغيرات اذ لا يحتجز سوى
، اذ VB4 من الاصدار 16bit وأتت معظم استخداماتها في نسخة VB4 منذ الاصدار
التي API تستخدم آثيرا عند الاتصال باجراءات Byte أتت المصفوفة من النوع
تتعامل مع الحروف، ام ا مع الاصدارات الاحدث فلن تتمكن من الاستفادة وتطبيق
UNICODE ، لان الترميز المتبع Byte الطرق القديمة على المتغيرات من النوع
. باختصار، لا ASCII بايت آترميز 1 بايت للحرف الواحد وليس 2 يستهلك مساحة
Byte عند استخدامك للمتغيرات من الن وع Strings تضع في ذهنك أي قضايا حرفية
، فيمكن قصر استخدامك لها على API خاصة عند الغوص في اعماق اجراءات
Byteالاعداد الصغيرة او البيانات الثنائية مع المتغيرات من نوع .

Integerالمتغيرات من النوع :

76

] للمتغيرات من النوع 32,767 ، 32,768اسند أي قيمة عددية صحيحة في المجال [-
الخاصة API بايت. وعند الحديث عن اجراءات 2 فهي تحجز مساحة Integer
. بعيدا عن UNICODE هي الانسب للترميز Integer بالحروف، فالمصفوفة من النوع
الحرفية، تفيدك المتغيرات من هذا النوع عند التعامل مع الاعداد API اجراءات
لقدرتها على احتواء Long الصحيحة، الا انني احبذ استخدام المتغيرات من النوع
، آما انها النوع القياسي لاغلب Integer قيم آبر بكثير من المتغيرات من النوع
. اما في حالة المصفوفات الكبيرة، فانني افضل استخدام المتغيرات API اجراءات
% من مساحة الذاآرة. 50 لتوفير Integer من النوع

Longالمتغيرات من النوع :

تستطيع حمل قيم عددية صحيحة في المجال [- Long المتغيرات من نوع
بايت للمتغير 4] فهي تحجز مساحة قدرها 2,147,483,647 ، 2,147,483,648
الواحد، وآما ذآرت في الفقرة السابقة اني احبذ استخدامها عوضا عن المتغيرات
، فهي تحمل قيم آبيرة جدا مقللة الخوف من ظهور خطأ وقت Integer من النوع
، فلو أكتبت آود يقرأ حجم ملف معين وآنت من المدمنين للنوع Overflow التنفيذ
، فستصاب بخيبة امل آبيرة عندما تتعامل مع الملفات التي تزيد احجامها Integer
: 32,767 عن

```
Dim iFileSize As Integer
`بايت 32,676 سيظهر خطأ اذا زاد حجم الملف عن
iFileSize =FileLen )"C:\MyFile.DAT"(
```

Booleanالمتغيرات من النوع :

ولكن القيم Integer هي نفس المتغيرات من النوع Boolean المتغيرات من النوع
، حجم المتغيرات من 1 True او - 0 False التي تمكنك من اسنادها اليها تكون اما
بايت، الا انها لا 2 أي Integer مثل حجم المتغيرات من النواع Boolean النوع

بايت يعتبر 2 بت الاخرى . صحيح ان الحجم 15 بت متجاهلة ال 1 تستخدم سوى
تسهل عليك عملية قراءة Boolean زيادة غير مستخدمة، الا ان المتغيرات من النوع
وفهم الآواد.
77
Singleالمتغيرات من النوع :
احتوائها هو الاعداد الموجبة Single مجال القيم التي يمكن للمتغيرات من النوع
الى 3.402823e38او الاعداد السالبة من - 3.402823e38 الى 1.401298e-45 من
بايت. 4 وتستهلك مساحة 1.401298e-45-
Double على النوع Single النوع Visual Basic ربما يفضل معظم مبرمجي
لاعتقادهم ان الأول اسرع في التنفيذ من الثاني، هذا الاعتقاد صحيح في النوع
Math Coprocessorالقديم من المعالجات والتي لا تحتوي على مساعد رياضي ،
اما اغلب المعالجات الجديد تحتوي على المساعد بغدادي وهو خاص بالعمليات
مما يجعل السرعة متقاربة Floating Point الحسابية للاعداد ذات الفاصلة العائمة
عوضا عن Double ، لذلك ينصح باستخدام النو ع Double و Single جدا بين النوعين
ودقة اعلى للاعداد لكبر مجال القيم OverFlow حتى تتقي شر الخطأ Single النوع
اسرع بكثير من Single الممكنة به ا. من ناحية اخرى، قد تكون المتغيرات من النوع
عند التعامل مع الخصائص او الطرق التي تحتك مع Double المتغيرات من النوع
ScaleHeight، ScaleWidth، Line، Circle، CurrentX الاحداثيات بشكل مل حوظ آ
معها ابطأ Double ، واستخدام النوع Single .... الخ فهذه الاحداثيات تستخدم النوع
Singleيضطر الى تحويل متغيرات النوع السابق الى . Visual Basic لان
Doubleمن النوع :____________**المتغيرات**
احتوائها هو الاعداد الموجبة Double مجال القيم التي يمكن للمتغيرات من النوع
او الاعداد السالبة 1.79769313486232e308 الى 4.9406564581247e-324 من
وتستهلك 1.79769313486232e308الى - 4.9406564581247e-324من -
بايت. 8 مساحة
لذ لك هو Double الخاصة بالاعداد تعود بقيمة من النوع Visual Basic معظم دوال
النوع المفضل دائما، الا ان عيبه الوحيد هو في المساحة الكبير التي يحتجزها، وقد
Doubleيظهر هذا العيب جليا في المصفوفات الكبيرة من النوع .
Currencyالمتغيرات من النوع :
Fixedالاحتفاظ بقيم عشرية للفاصلة الثابتة - Currency يمكن للمتغيرات من النوع
، 922,337,203,685,477.5808شريط ة ان تكون محصورة في داخل المجال [- Point
بايت ايض ا. يوفر هذا النوع من المتغيرات 8] وحجمها 922,337,203,685,477.5808
.... الخ والتي تستخدمها Fix، Round عناء التقريب باستخدام دوال التقريب آ
مما يبطئ العمليات الحسابية، مع Single و Double بكثرة مع المتغيرات من النوع
ابطأ خمس او اربع مرات من Currency ذلك الاستخدام المجرد للمتغيرات من النوع
78
فلا تستخدمها بكثرة في حالة تطبيق آلاف العمليات Single و Double المتغيرات
الحسابية عليها.
Decimalالمتغيرات من النوع :
آبيرة جدا، ولا يوجد Decimal الاعداد التي يمكنك اسنادها الى المتغير ات من النوع
على قيد الحياة . لن تستطيع تصريح MSDN داعي لذآرها هنا مادامت مكتبة
، وانما Dim X As Decimal مباشرة بالطريقة التقليدية Decimal المتغيرات من النوع
بايت- ومن ثم تسند قيمة له: 16 -الذي يستهلك Variant تستخدم النوع

```
Dim X As Variant
X =CDec )Text1.Text * (CDec )Text2.Text(
```

هي ابطأ انواع المتغيرات آما ستقرأ في Variant ولا تنسى ان المتغيرات من النوع
" قريبا. Varaint فقرة "المتغيرات من النوع
Dateالمتغيرات من النوع :

31 الى 100 ين اير 1 هذا النوع من المتغيرات يحمل قيم تاريخية تبدأ من التاريخ
ص حتى 00:00:00 ويشمل نفس المتغير وقت يبدأ من الساعة 9999 ديسمبر
بايت، وفي حقيقة الامر المتغيرات من 8 م وتستهلك مساحة 23:59:59 الساعة
، فالجزء العشري يمثل وقت Double هي نفس المتغيرات من النوع Date النوع
تمثل الساعة الثانية 37257.5 معين والجزء الصحيح يمثل تاريخ معين، فالقي مة
. السبب الذي جعلني اذآر تفاصيل المتغيرات 2002 يناير عام 1 عشر ظهرا من يوم
من هذا النوع هو اعطائك افكار مرنة تمكنك من اجراء عمليات آثيرة على قيم
التاريخ وهذه قطرات من محيط الامثلة:

```
Dim dDateVar As Date
dDateVar =Now
`اطبع التاريخ فقط
Print Int)dDateVar(
`اطبع الوقت فقط
Print CDate)dDateVar -Int)dDateVar((
`اضف اسبوع واحد
Print dDateVar +7
`يوم 30 احذف
```

79

```
Print dDateVar -30
`ساعات 6 احذف
Print dDateVar -0.75
```

وخصوصا العشر ية،**ملاحظة:** ان لم تكن دقيق في كتابة الاعداد المناسبة
فان نتائج العمليات السابقة لن تكون متوقعة، لذلك ينصح باستخدام
آما سيفصلها VBA و VB دوال الوقت والتاريخ المضمنة في مكتبات
لك الفصل القادم.

Stringالمتغيرات من النوع :

! اذا String سهلة؟ والجواب بسبب المتغيرات الحرفية من نوع BASIC لماذا لغة ال
فانسى آل شئ متعلق بقضية حجز المساحة في الذاآرة C آنت من مب رمجي
سواء أن ديناميكيا او ستاتيكيا باستخدام المصفوفات، او التحقق من طول النص
سطور لاسناد قيمة الى متغير حرفي، ف 6 او 3 وغيرها من الامور التي تتطلب
String هو المتكفل بهذه الامور تلقائيا بمجرد تصريح متغير من النوع Visual Basic
او اسناد قيم حرفية له.
تعتمد Strings بت - اصبحت المتغيرات الحرفية 32 نسخة عيار– VB4 منذ الاصدار
. بصفة عامة، يوجد نوعان من انواع المتغيرات ASCII وليس UNICODE ترميز
Fixed-length لك هما المتغيرات ثابتة الطول Visual Basic الحرفية يوفرهما
Variable-Lengthوالمتغيرة الطول .
المتغيرات ثابتة الطول هي متغيرات حرفية عدد حروفها محدد في اثناء تصريحها ولا
يمكن ان يتغير:

```
Dim FixedStr As String *12
sFixedStr = "ترآي العامري"
```

مما 12 ان يحمله هو FixedStr فالعدد الاقصى من الحروف التي يمكن للمتغير
2 يستهلك UNICODE لا تنسى ان–بايت 24يؤدي الى استهلاك مساحة قدرها
بايت للحرف الواحد . من عيوب المتغيرات ثابتة الطول هو عدم توافقها مع تقنية
لا تدعم هذا النوع من API الداخلية واجراءات VBA و VB ومعظم دوال مكتبات COM
المتغيرات، وحتى لو أن عدد حروف القيمة المسندة اقل من عدد الحروف
المصرحة، فان المسافات " " ستحل محل الخانات الفارغة، ولا يمكن لهذا النوع من
، آما لا يمكنه Public المتغيرات ان تكون مرئية على مستوى الوحدة من النوع
آيلوبايت، الا ان الميزة التي تظهر بها عند اسناد 64 حمل عدد من الحروف آبر من

80

القيم الحرفية لهذا المتغيرات فنتائجها تكون دائما اسرع من المتغيرات من النوع
لا يقوم باي عمليات احتجاز في الذاآرة Visual Basic المتغيرة الطول، وذلك لان
والتحقق من المساحة المتوفرة .... الخ.
من Component Object Model او برمجة الكائنات المكونة COM**ملاحظة:**
من Windows والتي تمكن تطبيقات OLE التقنيات المبنية على
الاتصال وتبادل البيانات فيما بينها، الفصلان الثاني عشر والثالث
" يختصان بهذه التقنية. COM عشر "برمجة المكونات
فهي باختصار تغطي على Variable-Length بالنسبة للمتغيرات المتغيرة الطول
10جميع عيوب النوع السابق، الا انها تحتجز مس احة تعادل ضعف عدد الحروف +
بايتات اضافية تحوي معلومات عن المتغير الحرفي آحجمه وغيرها من التفاصيل
عنك، والعدد الاقصى من الحروف التي يمكن حفظها Visual Basic التي يخفيها
جيجا بايت. 2 في هذا النوع يصل إلى

Objectالمتغيرات من النوع :

او بنوع فئات هي Object معظم المتغيرات التي تمثل أسنادات سواء صرحت بالنوع
Objectمتغيرات من النوع :

```
Dim X As Object
Dim Y As Form
Dim Z As Text
```

- حتى الوصول الى Object المتغيرات من النوع–اود ان اؤجل شرح تفاصيل الكائنات
الفصل الخامس "البرمجة أسنادية التوجه " وحتى ذلك الحين، لا تسند أسناد الى
أسناد
Setالا باستخدام العبارة :

```
Set X =New MyClass
Set Y =Form1
Set Z =Text1
Z = "اسناد قيمة خاصية وليس أسناد"
Z.Text = "اسناد قيمة خاصية وليس أسناد"
```

لا تشغل بالك آثيرا بالكود السابق، فالفصل الخامس قادم اليك.
81

Variantالمتغيرات من النوع :

وتعدلت بنيته التحتية منذ VB3 في الاصدار Variantظهرت المتغيرات من النوع
، ويستطيع حمل جميع انواع البيانات COM حتى تتوافق مع تقنية VB4 الاصدار
.... الخ. Long، Date، Stringالسابق ذآرها مثل:
بايت،البايت الاول يحدد نوع القيمة 16 هو Variant الحجم الذي يستهلكه المتغير
لا تستخدم الا في حالة آون القيمة 7 الى 2 الموجودة في المتغير، والبايتات من
فهي تمثل القيمة التي يحملها 15 الى 8 ، اما البايتات من Decimal من النوع
المتغير.
ليس فقط في امكانية اشتمالها Variant الميزة التي تتميز بها المتغيرات من نوع
على انواع مختلفة من البيانات بل واجراء العمليات الحسابية او المنطقية عليها،
باختبار نوع المتغيرات ومن ثم اجراء العملية الحسابية او Visual Basic حيث يقوم
المنطقية المناسبة لها:

```
Dim X As Variant
Dim Y As Variant
Dim Z As Variant
X =2000 ' Integer قيمة من النوع
Y =CLng)2000 (' Long قيمة من النوع
Z =X +Y ' Long قيمة من النوع
X =CDbl)2.5 (' Double قيمة من النوع
Z =X +Y ' Double قيمة من النوع
```

لا تحاول الاعتماد على الطرق السابقة بشكل استثنائي، فقد تعطيك نتائج غير
يؤدي الى Variant متوقعة، فمثلا استخدام معامل الجمع + مع متغيرين من النوع
جمعهما اذا أتت قيم عددية، اما الحرفية فتتم عملية الدمج بينهما آاستخدام
Visual ، واذا أن احد المتغيرين حرفي والاخر عددي فسيقوم& معامل الدمج
بمحاولة تحويل القيمة الحرفية الى عددية، وان لم يستطع فرسالة الخطأ Basic
سيكون لها نصيب في الظهور: Type Mismatch
Dim X As Variant
Dim Y As Variant
Dim Z As Variant
X =20

82

Y " =20"
Z =X +Y ' Z =40
X " =20"
Z =X +Y ' Z ="2020"
Print Z
X =20
Y " =abcd"
Z =X +Y 'رسالة خطأ
واعجبت بها آثيرا، فتذآر انها ابطأ انواع Variant اذا فتنت في المتغيرات من النوع
المتغيرات، فلا تحاول الاعتماد عليها الا عند الضرورة القصوى او عند الحاجة
Decimalلاستخدام المتغيرات من النوع .
باستخدام Variant تستطيع معرفة نوع القيمة الموجودة في المتغير من النوع
: الدالةVarType
Dim X As Variant
X =20
Print VarType)X (' Integer وهو النوع 2 تطبع
X " =20"
Print X ' String وهو النوع 20 تطبع
والتي Empty فان القيمة الابتدائية له هي Variant اذا لم تسند أي قيمة للمتغير
IsEmptyتستطيع اختبارها بالدالة :
Dim X As Variant
Print IsEmpty)X (' True
X " =20"
Print IsEmpty)X (' False
X =Empty
Print IsEmpty)X (' True
لا تعتبر قيمة خالية فهي قيمة Null ، لان Empty فهي لا تعني Null اما القيمة
DataBasesمعينة تستخدم في الغالب مع قواعد البيانات :

83

Dim X As Variant
X =Null
Print IsNull)X (' True
Print VarType)X (' Null وهو النوع 1 تطبع
، لكن لا تنسى Objects يمكن لها ان تحتوي أسنادات Variant والمتغيرات من النوع
عند اسناد قيمة أسناد الى متغير، واذا اردت Set استخدام الكلمة المحجوزة
فهي تعطي نوع قيمة الخاصية VarType الاستعلام عن نوع المتغير، فلا تستخدم
فهي تفي بالغرض المطلوب: IsObject الافتراضية للكائن، اما الدالة
Dim X As Variant
Set X = Text1

Print IsObject)X (' True
Text1.Text ' = "النص" X.Text
يمكن لها ان تحوي مصفوفات آما سيأتي في Variant اخيرا، المتغيرات من النوع
فيمكن احتضانها UDT فقرة "التركيبات والمصفوفات "، وبالنسبة للتركيبات من النوع
مصرح على Public شريطة ان يكون التركيب من النوع Variant في المتغيرات
Public ، أو على مستوى وحدة الفئات العامة Module مستوى الوحدة البرمجية
.Classes

الثوابت

ابسط انواع الثوابت هي الثوابت العددية والتي يمكنك آتابتها مباشرة بالنظام
او Hexadecimal للنظام الستعشري H& او باضافة البادئة Decimal العشري
للنظام الثماني: O& البادئة

' 15 جميع الاعداد التالية تساوي
Print 15
Print &HF
Print &O17

من الضروري ان انبه هنا بان جميع الاعداد المستخدمة في النظام الستعشري
والتي تكتبها في آوادك Octal والنظام الثما ني 2،.... E، F، 1، 0 آ Hexadecimal

84

بعد نهاية & مالم تضيف الرمز Integer اعداد من النوع Visual Basicتعتبر في نظر
، قد تكون جملتي السابقة ليست ذات اهمية آبيرة Long العدد فسيكون من النوع
عند معظم المبرمجين المبتدئين، لذلك علي ان اشد انتباههم بهذا المثال:

'ستعشري
' &HF000 =61440
Print &HF000 '- 4096 هنا تطبع
Print &HF000 & ' 61440 هنا تطبع
'ثماني
'&O170000 =61440
Print &O170000 '- 4096 هنا تطبع
Print &O170000 & ' 61440 هنا تطبع

، والتي يشترط آتابتها بين Strings بعد الثوابت العددية تأتي الثوابت الحرفية
علامتي التنص يص المزدوجة " و "، ولاستخدام علامة التنصيص " في نفس الثابت
الحرفي، آررها مرتين:

'مخرجات الكود التالي هي:
'ثابت حرفي
' 123"456
' "

Print"ثابت حرفي"
Print "123""456"
Print """"

فكرة الثوابت المسماة شبيهه بفكرة المتغيرات، ويكمن الفرق بينهما في أن قيم
الثوابت لايمكنك تعديلها وقت التنفيذ لانها قيم ليست موجودة بالذاآرة آقيم
المتغيرات، وانما يتم استبدال هذه الاسماء بقيمتها الفعلية في الكود اثناء عملية
للبرنامج. EXE ، فالثوابت تحفظ مباشرة في الملف التنفيذي Compiling الترجمة
Constتستطيع تعريف ثابت جديد باستخدام العبارة :

Const PI =3.14
Print PI

آما يفضل تعريف نوع الثابت لزيادة سرعة التعامل معه:

85

Const PI As Double =3.14
Const PROGRAMMER_NAME As String = "ترآي العامري"

Const SPECIAL_VALUE As Long =&H32FE&
ارجو ان تلتزم بالقيمة المناسبة عند تحديد نوع الثابت، فلا تسند قيمة عشرية مثلا، لان قيمة الثابت ستتغير ان لم تظهر رسالة Integer لثابت صحيح -آالنوع الخطأType Mismatch :
Const PI As Integer =3.14 ` القيمة ستكون 3
Const PI As Integer ="abc ` ستظهر رسالة خطأ
على مستوى الاجراء المحلي، Private اخيرا، قابلية الرؤية الافتراضية للثوابت تكون او على مستوى نافذة النموذج او الفئة اذا صرح عنها في منطقة الاعلانات العامة، . مع تضمين BAS او على مستوى المشروع اذا صرح عنها في ملفات البرمجة الكلمة المحجوزةPublic :
Public Const PI As Double =3.14

الترآيبات والمصفوفات

بالاضافة الى انواع المتغيرات السابقة، تستطيع تخصيص انواع جديدة تعرف بالترآيبات، آما يمكنك ربط سلسلة من المتغيرات في مصفوفات احادية او متعددة الابعاد.

ترآيباتEnum

يمكنك تعريف نوع جديد من المتغيرات بحيث يحتوي على قيمة من مجموعة قيم لتعريف Enum . تستطيع استخدام الكلمة المحجوزة Enumeration تعرف بال الترآيب شريطة ان يكون في منطقة الاعلانات العامة، هذا مثال يعرف ترآيب لأيام الاسبوع:

```
Private Enum enmDay
Saturday
SunDay
MonDay
```



```
TuesDay
Wednessday
Thursday
Friday
End Enum
```

والان يمكنك استخدام الترآيب السابق لتعريف انواع جديدة من المتغيرات:

```
Dim X As enmDay
Dim Y As enmDay
X =Saturday
Y =X
```

او حتى استخدامها لاستقبال المتغيرات في اعلى الاجراءات:

```
Private Sub MySub)TheDay As enmDay(
If TheDay =Friday Then
"اجازة"MsgBox
Exit Sub
End If
End Sub
```

ماهي الا متغيرات عددية من النوعEnum حقيقة المتغيرات من النوع ترآيبات فتستطيع التعامل معها آما لو أنت متغيرات عددية: Long

```
Dim X As enmDay
X =Saturday
Print X
X =X +1
Print X
```

، ولتخصيص قيم من عندك، ارجع0 آما تلاحظ، يبدأ ترقيم عناصر الترآيب من العدد
الى تعريف الترآيب وضع القيم من عندك:

87

```
Private Enum enmDay
Saturday =20
SunDay =30
MonDay
TuesDay
Wednessday
Thursday
Friday
End Enum
```

1. مع العلم ان مقدار الزيادة لباقي العناصر تكون

ترآيبات من النوع UDT

User Defined يعرف هذا النوع من الترآيبات بالانواع المعرفة من قبل المستخدم
حيث تمكنك هذه الترآيبات من الاحتواء على انواع مختلفة من البيانات، Types
لتعريف ترآيب جديد: استخدم الكلمة المحجوزة Type

```
Private Type typPerson
sName As String
bSingle As Boolean
iAge As Integer
End Type
```

ويمكنك استخدامه مباشرة آما في هذا الكود:

```
Dim Turki As typPerson
Dim Ali As typPerson
Turki.sName = "ترآي العامري"
Turki.iAge =99
Turki.bSingle =True
Ali.sName = "علي العلي"
Ali.iAge =35
Ali.bSingle =False
```

88

```
Ali =Turki
Print Ali.sName
```

تفي بالغرض: LenB ولمعرفة حجم الترآيب، فالدالة

```
Print LenB )Turki(
```

لا تنسى انه يمكن للترآيبات ان تحتوي على ترآيبات اخرى:

```
Private Type typAdress
sCountrey As String
sCity As String
End Type
Private Type typPerson
sName As String
bSingle As Boolean
iAge As Integer
Address As typAdress
End Type
```

الوصول الى عناصر الترآيب المحضن يتم من خلال الترآيب الحاضن لها:

```
Dim Turki As typPerson
Turki.sName = "ترآي العامري"
Turki.iAge =99
```

Turki.bSingle =True
= "الظهران"Turki.Address.sCity
= "المملكة العربية السعودية"Turki.Address.sCountrey
بالنسبة لقابلية الرؤية فلن تستطيع تعريف التراآيبات باستخدام الكلمة المحجوزة
، واذا آنت عنيدا واصررت على استخدام الكلمة Classes الا في الفئات Public
بتصريح Visual Basic ، فسيسمح لك BAS في ملفات البرمجة Public المحجوزة
89
متغيرات جديدة من التراآيب العام، ولكنك ستصاب بخيبة امل آبيرة ان علمت ان
لا يمكنك استخدام هذه التراآيبات Public الاجراءات المصرحة على مستوى الوحدة
لها: Parameters آقيم مستقبلة
Public Sub MySub)P As typPerson ('غير ممكن
Private Sub MySub)P As typPerson ('ممكن
Freind Sub MySub)P As typPerson ('ممكن
، تستطيع تعريف التراآيبات على مستوى Classes **ملاحظة:** بالنسبة للفئات
1لا تساوي - Instancing شريطة ان تكون قيمة الخاصية Public
.Private

المصفوفات

سواء أنت احادية Arrays من انشاء والتعامل مع المصفوفات Visual Basic يمكنك
بعدا: 60 قد تصل الى–البعد او متعددة الابعاد
Dim OneDim)99 (As Intger 'عنصر 100
Dim TwoDim)4, 9 (As Integer 'ثنائية الابعاد
Dim ThreeDim)2, 2, 2 (As Integer 'ثلاثية الابعاد
Dim OneDArray)0 To 10 (As String
Dim TwoDArray)0 To 10, 0 To 10 (As Long
Dim OneDArray)15 To 22 (As String
تستطيع البدء في عملية اسناد القيم بمجرد تصريح المصفوفة مع العلم ان فهرس
Option يبدأ من صفر مالم تستخدم الكلمة المحجوزة Array Index المصفوفة
في منطقة الاعلانات العامة للوحدة البرمجية فانه سيبدأ بواحد: Base 1
OneDim)0 = (100
OneDim)1 = (200
TwoDim)0, 0) = (100, OneDim)0 + (OneDim)1((
90
ملاحظة: رغم ان بدء ترقيم فهرس المصفوفة يمكن ان يبدأ بواحد، الا انني لا
فعل ذلك، فعند نقل الآواد بين Visual Basic احبذ لمبرمجي
المشاريع المختلفة او الوحدات البرمجية المختلفة قد لا يتم تفعيل
مما يترتب عنه ظهور عشرات Option Base 1 الكلمة المحجوزة
الاخطاء البرمجية.
تعود برقم UBound بينما الدالة LBound ولمعرفة رقم العنصر الاول استخدم الدالة
العنصر الاخير:
Dim lCounter As Long
For lCounter =LBound)OneDim (To UBound)OneDim(
Print OneDim)lCounter(
Next
و UBound اما مع المصفوفات المتعددة الابعاد، عليك ارسال رقم البعد مع الدالتين
:LBound
Print UBound)TwoDim (' 4 تطبع
Print UBound)TwoDim, 1 (' 4 تطبع
Print UBound)TwoDim, 2 (' 9 تطبع
هي مصفوفات ستاتيكية أي ThreeDim و OneDim، TwoDim المصفوفات السابقة

ثابتة الحجم لا تتغير في وقت التنفيذ، لذلك فالمرونة الحقيقة ستكون مع
التي تتيح لك التحكم في حجم Dynamic Arrays المصفوفات الديناميكية
المصفوفات آلما دعت الحاجة، وتصريحها يكون بدون ذآر حجمها:
Dim DynamicArray ()As String
اولا ReDim قبل ان تبدأ في عملية اسناد القيم، عليك استخدام الكلمة المحجوزة
مع ذآر الحجم:
ReDim DynamicArray)2(
"نوره" (= DynamicArray)0
91
"العنود" (= DynamicArray)1
"الهنوف" (= DynamicArray)2
مرة اخرى وعليك معرفة ReDim لو اردت زيادة او تقليص حجم المصفوفة، استخدم
ان جميع محتويات المصفوفة سوف تلغى:
ReDim DynamicArray)4(
"جنان" (= DynamicArray)3
"زغلول!" (= DynamicArray)4
تطبع "زغلول!"' (Print DynamicArray)4
لا تطبع شيئ' (Print DynamicArray)2
واذا رغبت بتغيير حجم المصفوفة دون المخاطرة بفقد البيانات الموجودة فيه ا،
جاهزة للاستخدام: Preserve فالكلمة المحجوزة
ReDim Preserve DynamicArray)4(
"جنان" (= DynamicArray)3
"زغلول!" (= DynamicArray)4
تطبع "زغلول!"' (Print DynamicArray)4
تطبع "الهنوف"' (Print DynamicArray)2
يقودني لاخبارك انك لن تستطيع تغيير ابعاد المصفوفة، Preserve الحديث عن
فالمصفوفات الديناميكية التالية:
Dim OneDim ()As Integer
Dim TwoDim ()As Integer
ReDim OneDim)4(
ReDim TwoDim)2, 2(
Preserveلن تستطيع تغيير ابعادها باستخدام :
92
'مستحيل
ReDim Preserve OneDim)3, 3(
ReDim Preserve TwoDim)1(
'ولكن هذا ممكن
ReDim OneDim)3, 3(
ReDim TwoDim)1(
للمصفوفات الديناميكية هي امكانية VB6 من المزايا التي اضيفت الى الاصدار
نسخ قيم مصفوفة آاملة الى اخ رى في سطر واحد شريطة ان تكونا من نفس
النوع، فبامكانك كتابة شيئا مثل:
Dim MyArray)20 (As Integer
Dim YourArray ()As Integer
MyArray)0 = (10
MyArray)1 = (20
...
YourArray = ()MyArray ()
Print YourArray)0 (' = 10
نقطة اخيرة حول المصفوفات الديناميكة وهي امكانية تدميرها باستخدام العبارة

:Erase
Erase OneDim
مرة اخرى: **Variant**النوع
يمكن لها ان تمثل مصفوفات اما عن طريق اسناد Variant المتغيرات من النوع
مصفوفة لها:
Dim Cities)2 (As String
Dim vCities As Variant
Cities)0"بغداد" = (
Cities)1"جدة" = (
Cities)2"ابها" = (
vCities =Cities
Print vCities)1 (`جدة
93
Arrayاو باستخدام الدالة :
Dim vCities As Variant
vCities =Array" بغداد") ," جدة" ,("ابها"
Print vCities)0 (`بغداد
هو انشاء مصفوفات Variant من الاشياء التي تعجبني في المتغيرات من النوع
، وهي مصفوفات ابعادها تختلف من Variable-Dimension Arrays مختلفة الابعاد
عنصر لاخر، فقد يكون العنصر الاول احادي البعد والثاني ثنائي البعد والثالث ثلاثي
على Variant البعد وتطبيقها سهل جدا، الطبخة آلها احتواء مصفوفة من النوع
عدة مصفوفات:
Dim VarDim)2 (As Variant
VarDim)0"احادي البعد" = (
VarDim)1 = (Array")1 "ثنائي البعد," 2 ("ثنائي البعد
VarDim)2 = (Array")1 "ثلاثي البعد," 2 "ثلاثي البعد," 3 ("ثلاثي البعد
Print VarDim)0(
Print VarDim)1)(0(, VarDim)1)(1(
Print VarDim)2)(0(, VarDim)2)(1(, VarDim)2)(2(

المجموعات

تلعب المصفو فات دورا حيويا في برامجك الجدية خاصة بالامور التي تتعلق بالحلقات
تعتبر آثر مرونة من المصفوفات Collections التكرارية وغيره ا. الا ان المجموعات
من حيث اضافة وازالة العناصر منه ا. الهدف من هذه الفقرة هو تعريفك بالمجموعات
وطرق استخداماتها.

مزايا المجموعات:

، وظيفتها VBA عبارة عن أسنادات مشتقة من مكتبة Collections المجموعات
الرئيسية مثل وظيفة المصفوفات تماما الى انها تختلف عنها في النقاط التالية:
94
- المجموعات لا تحتاج الى تحديد حجمها عند عملية تصريحه ا. فعند تصريحك
لمجموعة جديدة تستطيع اضافة العناصر لها وقت التنف يذ ديناميكيا أي بدون تحديد
أي حجم لها ستاتيكيا.
- تستطيع اضافة العناصر الى المجموعات في أي مكان تريده في المجموعة . أي
لست مضطر ا لوضعها في نهاية القائمة مثل المصفوفات، فالمجموعات تعطيك حرية
آبيرة في اضافة وحذف العناصر سواء أتت في بداية المجموعة او نهايته ا او حتى
في وسطها.
- المجموعة الواحدة يمكن ان تحتوي عناصرها على انواع بيانات مختلفة، أي قد
بعكس المصفوفات التي لابد من Integer والثاني String يكون العنصر الاول فيها
توحيد نوع عناصرها.
بدون Key - توفر المجموعات طرق اخرى لتحديد عناصرها عن طريق مفاتيح

استخدام اسلوب الترقيم الذي تتبعه المصفوفات.
بعد هذه المزايا قد تحرم استخدام المصفوفات وتنتقل الى المجموعات الى الابد !
لكن من المهم جدا ان تضع في ذهنك ان المجموعات ابطأ من المصفوفات بآثر من
مرة ! لذلك، لا تستخدمها ان أتت السرعة تعني لك الشئ الكثير . المزيد 100
ايضا، المجموعات تستهلك مساحة اآبر بكثير من المساحة المطلوبة في الذاآرة ،
ويعيب المجموعات ان عناصرها المضافة قابلة للقراءة لكن ليست قابلة للتعديل !
فلتعديل قيمة عنصر من عناصر المجموعة، عليك القيام باسلوب غير مباشر آحذف
العنصر المراد تعديله واضافته من جديد بعد التعديل.

برمجة المجموعات:

الخطوة الاولى التي تحتاجها هي انشاء أسناد المجموعة والذي تحصل عليه من
Collectionالنوع -الفئة- :

```
Dim MyCol As New Collection
```

مع ارسال قيمة العنصر له ا. Add ولاضافة عناصر الى المجموعة، استخدم الطريقة
اذ ا اردت حتى تستخدم آطريقة اخرى للوصول Key آذلك، تستطيع ارسال مفتاح
الى العنصر بدون الاعتماد على رقمه:

95

```
MyCol.Add "amazon.com", "shopping"
MyCol.Add "hotmail.com", "mail"
MyCol.Add "yahoo.com", "Search"
```

-او حتى تجاه لا- Item تستطيع الوصول الى عناصر المجموعة عن طريق الطريقة
وارسال رقم العنصر او مفتاحه:

```
Print MyCol.Item)1 (' amazon.com
Print MyCol.Item")mail ("' hotmail.com
Print MyCol")Search ("' yahoo.com
```

Removeولحذف العنصر استخدم الطريقة :

```
MyCol.Remove 1
MyCol.Remove "mail"
```

واسرع طريقة تمكنك من حذف جميع العناصر، قم بانهاء أسناد المجموعة:

```
Set MyCol  =Nothing
```

لكن تذآر! عليك باعادة انشاء الكائن حتى تتمكن من اضافة عناصر جديدة:

```
'خطأ في هذا السطر
MyCol.Add "amazon.com", "shopping"
'لابد من انشاء الكائن من جديد
Set MyCol  =New Collection
MyCol.Add "amazon.com",  "shopping"
```

Countاخيرا، لمعرفة عدد العناصر في المجموعة استخدم الطريقة :

```
Print MyCol.Count
```

96

الاجراءات والدوال

Functionsودوال ، Sub's من تعريف اجراءات Visual Basic يمكنك حبيب القلب
حيث يمكن للدوال من العودة بقيمة بعد نهاية تنفيذ الدالة، نوع القيمة التي تعود
بها الدالة هو النوع الذي تكتبته في نهاية تعريف الدالة:

```
'دالة تعود بقيمة حرفية
Function GetUserName  ()As String
GetUserName = "ترآي العامري"
End Sub
'دالة تعود بترآيب UDT
Function GetPersonData  ()As typPerson
GetPersonData.sName = "ترآي العامري"
```

GetPersonData.iAge =99
End Function

الكود الموجود داخل الدالة، يتعامل مع اسم الدالة آمتغير من نفس نوع الدالة:

```
Function OddNumbers ()As String
Dim iCounter As Integer
OddNumbers ="”
For iCounter =0 To 9
If iCounter Mod 2 <> 0 Then
OddNumbers =OddNumbers & iCounter
End If
Next
End Function
```

تنتهي عملية تنفيذ الاجراء او الدالة بمجرد الوصول الى نهايته، تستطيع انهاء Exit Function للاجراءات والعبارة Exit Sub تنفيذهما قبل ذلك باستخدام العبارة لانهاء Exit Function يستخدم Recursion للدوال، هذا مثال للخوارزميات التراجعية عملية تنفيذ الدالة:

97

```
Function Factorial)iNum As Integer (As Long
If iNum =1 Then
Factorial =1
Exit Function
Else
Factorial =iNum *Factorial)iNum -1(
End If
End Function
```

، أي تستطيع Public بالنسبة لقابلية الرؤية للاجراءات والدوال فالافتراضية استدعائها من أي مكان في المشروع:

```
BAS في ملف برمجة‘
Sub MySub1 ()
...
End Sub
Form1 في نافذة نموذج‘
Public Sub MySub2 ()‘ Public يمكنك تجاهل الكلمة المحجوزة
...
End Sub
Form2 في نافذة النموذج‘
Private Sub Form_Load()
MySub1
Form1.MySub2
End Sub
```

السابقة لاسم الاجراء او الدالة، فهي تمنع المبرمج Private اما الكلمة المحجوزة من استدعاء الاجراء من خارج الوحدة البرمجية:

```
Form1 في نافذة نموذج‘
Private Sub MySub ()
...
End Sub
```

98

```
Form2 في نافذة النموذج‘
Private Sub Form_Load()
Form1.MySub ‘فقط Form1 لن تستطيع استدعاء الاجراء لانه خاص بالنموذج
End Sub
```

في Public فهو نفس تأثير الكلمة المحجوزة Friend اما تأثير الكلمة المحجوزة ، ويكمن الفرق بينهما في حالات بناء مشاريع Standard EXE المشاريع القياسية يمكن ان يستدعيها المبرمج Public حيث ان الاجراءات من النوع ActiveX من النوع فلن Friend ، اما الاجراءات من النوع COM من خارج المشروع عن طريق الاتصال يتمكن أي شخص من استدعائها ما دامت آاواده خارج آاواد مشروعك.

الارسال بالمرجع او القيمة

. تحديد Parameters قابلة لاستقبال متغيرات Functions او Sub's الاجراءات المتغيرات التي يحتاجها الاجراء يعتمد بشكل آبير على الهدف والوظيفة التي يقوم يمكن ان Functions او Sub's بها الاجراء . بصفة عامة، الاجراءات سواء أتت String او Integerتستقبل انواع مختلفة من المتغيرات سواء أتت انواع قياسية آ Classesاو فئات . UDT .... الخ، او حتى انواع معرفة من قبل المستخدم آترآيبات . عملية ارسال المتغير Arrays بالاضافة الى قابلية الاجراءات لاستقبال المصفوفات الى الاجراء تتطلب وضع القيم بين قوسين في حالة آون الاجراء سيعود بقيمة لمستدعي ذلك الاجراء، اما غير ذلك فلا يشترط وضع الاقواس.

افتراضيا، تستقبل الاجراءات المتغيرات المرسلة لها بالمرجع . وان رغبت في جعلها عند تصريح المتغير الذي ByVal تستقبل المتغيرات بالقيمة فلابد من كتابة الكلمة يستقبله الاجراء . اما الفرق بين عملية ارسال المتغير بالمرجع وارساله بالقيمة فهو تقوم على اساس ارسال Reference بسيط جد ا. فعملية ارسال المتغير بالمرجع مؤشر للمتغير أي عنوان المتغير بالذاآرة والذي عن طريقه تستطيع تعديل قيمة المتغير من الاجراء:

```
Sub Start()
`هنا نقطة البداية
Dim X As Integer
' X  =0
Print X
MySub X
' X  =5
99
Print X
End Sub
Sub MySub )X As Integer(
X  =5
End Sub
```

توقع استقبال مرجع لمتغير من نوع MySub فكما تلاحظ في الكود السابق، الاجراء مما يعطيه القدرة على تغيير القيمة المرسلة له. الكثير من الحلول يوفرها Integer لك اسلوب الاستدعاء بالمرجع لعل اسهلها اجراء يقوم بتبديل قيمتين مرسلتين له:

```
Sub Swap )X As Variant, Y As Variant(
`اجراء يستبدل قيم انواع مختلفة
`من المتغيرات المرسلة
Dim vTemp As Variant
vTemp  =X  :X  =Y  :Y  =vTemp
End Sub
```

Strings اما الارسال بالقيمة فهو ابطأ من الارسال بالمرج ع خاصة مع الحروف سيضطر لنسخ قيمة المتغير الى مكان مؤقت Visual Basic والسبب في ذلك ان في الذاآرة، آذلك لن تستطيع تعديل قيمة المتغير المرسل لك . لجعل الاجراء ByValيستقبل متغيرات بالقيمة استخدم الكلمة المحجوزة :

```
Sub Start()
`هنا نقطة البداية
Dim X As Integer, Y As Integer
```

```
' X  =0, Y  =0
Print X, Y
MySub X, Y
' X  =3, Y  =0
Print X, Y
End Sub
Sub MySub )X As Integer, ByVal Y As Integer(
X  =5
```

100

```
Y  =10
End Sub
```

ارسال انواع اخرى

.... الخ يمكن ان String، Integer، Double جميع الانواع القياسية للمتغيرات
، أسنادات مكتبات Controls ترسلها وتستقبلها الاجراءات، اما الانواع الاخرى آالادوات
فتحتاج لشئ من UDT .... الخ والترآيبات App، Printer، Screen آ VBA و VB
التفصيل:

ارسال متغيرات الكائنات:

فيمكنك ارسال متغيرات تابعة لكائنات Standard EXE بالنسبة للمشاريع القياسية
ActiveX اللغة الى الاجراءات دون أي مشاآل، اما بالنسبة للمشاريع الاخرى آ
.... الخ فلا يمكنك عمل ذلك الا للاجراءت التابعة لمشروعك EXE، ActiveX OCX
، والسبب في ذلك Friend او Private وهي الاجراءات التي تصرح بالكلمة المحجوزة
يمكن ان تستدعى من برامج Public يبدو بديهي ا اذا علمت ان الاجراءات من النوع
COM آما ستعرف في الفصل الثاني عشر "برمجة المكونات COM اخرى عن طريق
".1

UDTارسال المتغيرات الترآيبات :

Type والتي تنشئها عن طريق الكلمة المحجوزة UDT المتغيرات من نوع الترآيبات
تستطيع ارسالها لكن في حالات خاصة وشروط معينة اختصرها في هذه النقاط:
ومعرف في داخل وحدة برمجية آنافذة Private - اذا أن مجال الترآيب من نوع
نموذج، فئة، ملف برمجة الخ .. فان الاجراءات التي داخل تلك الوحدة البرمجية يمكن
لها ان تستقبل المتغيرات من نوع ذلك الترآيب.
فان جميع BAS معرف في ملف برمجة Public - اذا أن الترآيب مجاله عام
الاجراءات التي في نفس المشروع قابلة لاستقبال ذلك الترآيب شريطة ان لا تكون
Publicهذه الاجراءات من النوع .
، فان ActiveX في مشروع من النوع Class - في حالة آون الترآيب معرف في فئة
الاجراءات الموجودة في بر نامجك تستطيع ان تستقبل ذلك الترآيب المعرف في
. 1-Private لا تساوي Instancing اذا أنت الخاصية COM داخل مكون

101

تخصيص المتغيرات المرسلة

او غير محدودة Optional تستطيع تخصيص المتغيرات المرسلة اما بجعلها اختيارية
العددUnlimited Parameters .

المتغيرات الاختيارية:

اذا أن عدد المتغيرات المرسلة للاجراء غير عدد المتغيرات المصرحة فيه والتي
يتوقعها، فان رسالة الخطأ لها نصيب . لذلك، في حالات آثيرة جدا تريد ان تجعل
اجراءك مرن بما فيه الكفاية وذلك عن طريق تعريف متغيرات اختيارية تعطي
المستخدم حرية ما اذا أن يريد ان يرسلها او لا . تتم هذه العملية عن طريق
. واذا اردت معرفة ما اذا أن المستخدم قد ارسل قيمة Optional الكلمة المحجوزة
تعود IsMissing ، لكن احذر ! دالة IsMissing الى الاجراء تحقق عن طريق الدالة
فقط: Variant بقيم صحيحة في حالة آون المتغير المرسل لها من نوع

```
Sub Start()
`هنا نقطة البداية
' MyFunction - =1
Print MyFunction()
' MyFunction  =4
Print MyFunction )2(
End Sub
Function MyFunction )Optional X As Variant (As Integer
If IsMissing)X (Then `لم يتم ارسال قيمة للمتغير
MyFunction - =1
Else `تم ارسال قيمة
MyFunction  =X ^ 2
End If
End Function
```

حتى نتمكن التحقق من Variant المشكلة هنا ان المتغيرات لابد ان تكون من نوع وهو ابطأ الانواع استخدام . لذلك، يبدو ان الحل IsMissing ارسالها عن طريق الدالة الافضل هو عن طريق وضع قيمة افتراضية تستخ دم في حالة لم يتم ارسال قيم للمتغيرات:

102

```
Sub Start()
`هنا نقطة البداية
' MyFunction - =1
Print MyFunction()
' MyFunction  =4
Print MyFunction )2(
End Sub
Function MyFunction )Optional X As Integer - =1 (As Integer
If X - =1 Then `لم يتم ارسال قيمة للمتغير
MyFunction - =1
Else `تم ارسال قيمة للمتغير
MyFunction  =X ^ 2
End If
End Function
```

لابد من ان تكون في نهاية سلسة Optional **ملاحظة:** المتغيرات الاختيارية المتغيرات المرسلة الى الاجراء -أي من جهة اليمين.

متغيرات غير محددة العدد:

في هذه الحالة، فانك لا تحدد عددا معينا من المتغيرات التي سيستقبلها الاجراء لان القيم ستكون في مصفوفة تعرفها بنفسك عن طريق استخدام الكلمة . هذا مثال لدالة Variant ، شريطة ان يكون المتغير من نوع ParamArray المحجوزة توجد مجموع القيم المرسلة لها:

```
Function Sum )ParamArray args ()As Variant (As Long
Dim iCounter As Integer
For iCounter  =0 To Ubound)args(
Sum = Sum  +args)iCounter(
Next iCounter
End Function
```

ولاستدعاء الدالة آل هذه الامثلة صحيحة:

103

```
'Sum  =10
Print Sum )5, 5(
'Sum  =100
```

Print Sum)20, 20, 20, 20, 20(
'Sum =1000
Print Sum)250, 250, 250, 250(

التحكم في سير البرنامج

من الاجراءات لن تكون ذات قيمة معنوية آبيرة مالم تستخدم عبارات التفرع 90%
لتتحكم في سير Do ... Loop او For ... Next او الحلقات التكرارية آ Select و If آ
وعبارات التكرار Branch Statements البرنامج، الفقرات التالية تشرح عبارات التفرع
.Looping Statements

التفرع باستخدامIF

وحسب وانما Visual Basic الجميلة لا يستغني عنها اي مبرمج، ليس في If جملة
من آثر العبارات استخدام ا في If في جميع لغات البرمجة . ومما لا شك فيه تعتبر
المفضل- عدة سطور:–البرنامج، وهي تنجز اما في سطر واحد او

```
`في سطر واحد
If X > 0 Then Y =0
If X > 0 Then Y =0 Else Y =X
If X > 0 Then X =0 :Y =0 Else Y =X
`في عدة سطور
If X > 0 Then
Y =0
End If
If M > 0 Then
T =1
Else
T - =1
End If
If M > 0 Then
```



```
T =1
ElseIf M < 0 Then
T - =1
Else
T =0
End If
```

اختصار الجمل الشرطية:

Trueهو ، If اذا تحقق الشرط او اصبحت نتيجة التعبير الشرطي الذي يلي جملة
سيتم تنفيذها: If ... Then فان آلواد التي تلي عبارة

```
If Value =True Then
End If
If x <> 0 Then
End If
```

الاولى اننا اختبرنا If تستطيع اختصار الجمل الشرطية، فلو لاحظ ت في جملة
اذ ا م ا أتت صح او خط أ. المبرمجون المتمرسون يفضلون Value القيمة المنطقية
آتابتها بهذا الشكل:

```
If Value Then
End If
```

يعتبر اي قيمة غير الصفر Visual Basic الكود السابق صحيح والسبب في ذلك، ان
. قد تختصر If في حالة اختبار الشروط مع جملة False اما الصفر فهي True هي
جملة الشرط الثانية ايضا آما في هذا الكود:

```
If x Then
```

```
End If
```

105

صحيح ان النتيجة مماثلة، لكن حاول تجنب الاختصار في حالة استخدام قيم غير منطقية حتى لا تظهر لك نتائج غير متوقعة . لان القيم المناسبة للاختصارات ه ي القيم المنطقية فقط وفيما عدى ذلك قد يسبب لك الكثير من الشوائب . راقب الكود التالي:

```
x =3
y =4
If x <> 0 And y <> 0 Then 'الطريقة الصحيحة
If x And y Then 'مشكلة في هذا الاختصار
```

لقد استخدمت اسلوب الاختصار ويبدو ان جملتى الش رط متماثلتين، لكن هل تصدق ان الشرطين السابق ين مختلفان تمام ا! اذا اردت معرفة السبب قم بالرجوع وستعرف السبب ، 0100 و 0011 بنظام الاعداد الثنائي y و x الى قيمة المتغيرين تقوم بمقارنة Or و And والذي نستنتج منه باختصار على ان معاملات الربط آ القيم الثنائية للعدد بغض النظر عن نوعه.

التفرع باستخدام Select

بامكانك تطبيق مبدأ التفرع باستخدام عبارة Select :

```
Select Case iDay
Case 1
sDay = "السبت"
Case 2
sDay = "الاحد"
...
Case 7
sDay = "الجمعة"
Case Else
sDay = "غير معرف"
End Select
```

تكمن قوة عبارة Case في امكانية تطبيق المعاملات المنطقية او تحديد مجال للقيم:

106

```
Select Case iAge
Case Is < =0
sDesc = "لم يولد"
Case 1 To 11
sDesc = "طفل"
Case 15 To 20
sDesc = "مراهق"
Case 21 To 50
sDesc = "رجل"
Case Is > =51
sDesc = "شايب"
End Select
```

المزيد ايضا، يمكنك تحديد مجموعة قيم:

```
Select Case sLetter
Case "A" To "B", "a" To "b"
sLetter = "حرف ابجدي"
Case "0" To "9"
sLetter = "عدد"
Case ".", ":", " ", ";", "?"
```

"رمز" = sLetter
Case
"غير معروف" = sLetter
End Select

ليس هذا فقط، بل يمكنك تحديد مجموعة جمل شرطية:

Select Case True
Case x > 0, Y < 0
`تعادل
` If)X > 0 (Or)Y < 0(
End Select
Select Case False
Case x > 0, Y < 0
`تعادل

107

` If)Not)X > 0 ((Or)Not)Y < 0((
End Select

الحلقات التكرارية

For ... Nextحدد القيمة الابتدائية والقيمة النهائية للحلقة :

Dim iCounter As Integer
For iCounter =2 To 4
Print iCounter `سيتكرر تنفيذ الامر ثلاث مرات
Next

وعليك معرفة ان مقدار الزيادة سيضاف الى متغير الحلقة حتى بعد نهايتها:

Dim iCounter As Integer
For iCounter =1 To 3
Print iCounter
Next
Print iCounter ` 3 وليس 4 قيمة المتغير بعد نهاية الحلقة

Stepتستطيع التحكم في مقدار الزيادة او النقصان باستخدام :

Dim iCounter As Integer
For iCounter =10 To 0 Step -1
Print iCounter
Next

آهذا الكود الذي يطبع جدول Nested Loops يمكنك تطبيق فكرة الحلقات المتداخلة الضرب:

Dim A As Integer
Dim b As Integer

108

For A =1 To 5
For b =A To 5
Print A, "x", b, "=", A *b
Next
Next

Exit Forبامكانك انهاء الحلقة في أي وقت تريد باستخدام العبارة :

Dim iCounter As Integer
For iCounter =0 To 100
If MsgBox"هل تريد انهاء الحلقة؟") , vbYes Then) = vbYesNo
Exit For
End If
...
Next

Collectionsفهي تطبق على أسنادات المجموعات : اما حلقة For Each

```
Dim ctrl As Control
`محاذاة جميع الادوات الى اليسار
For Each ctrl In Controls
ctrl.Left =0
Next
```

Variantاو حتى المصفوفات شريطة ان يكون متغير الحلقة من النوع :

```
Dim X)100 (As Integer
Dim Y As Variant
`آود لاسناد قيم للمصفوفة
...
`طباعة محتوياتها
For Each Y In X
Print Y
Next
```

109

، لانك لا For ... Next فهي آثر مرونة من الحلقة Do ... Loop اما بالنسبة للحلقة Untilاو : While تحدد عدد معين من التكرار وانما جملة شرطية باستخدام

```
Do While MsgBox"هل تريد انهاء الحلقة؟") , vbYes) = vbYesNo
...
Loop
Do Until MsgBox"هل تريد انهاء الحلقة؟") , vbNo) = vbYesNo
...
Loop
```

في حال True ستتم عملية تنفيذ الحلقة مادامت الجملة الشريط صحيحة في حال استخدام الكلمة المحجوزة False او While استخدام الكلمة المحجوزة . واذا اردت تنفيذ الحلقة التكرارية مرة واحد على الاقل، ضع حمل الشرط في Until اسفل الحلقة:

```
Do
...
Loop While MsgBox"هل تريد انهاء الحلقة؟") , vbNo ) = vbYesNo
Do
...
Loop Until MsgBox"هل تريد انهاء الحلقة؟") , vbYes) = vbYesNo
```

Selectاو ، If بامكانك وضع جملة الشرط في داخل الحلقة ايضا باستخدام عبارة Exit Doلكن لا تنسى انهاء الحلقة بالعبارة :

```
Do
If MsgBox"هل تريد انهاء الحلقة؟") , vbYes Then) = vbYesNo
Exit Do
End If
...
Loop
```

110

التحويل بين For ... Next : وDo ... Loop

والعكس صحيح، لكن Do ... Loop الى حلقة For ... Next تستطيع تحويل حلقة تمثل عدد For ... Next عليك الانتباه الى ان القيم التي تحددها في بداية الحلقة التكرار حتى وان تغيرت، فبالرغم من ان الحلقتين التاليتين متشابهتين:

```
A =5
For iCounter =1 To A
...
Next
```

```
iCounter =1
Do
...
iCounter =iCounter +1
Loop Until iCounter > A
```

For ...، فالحلقة A الا ان الاختلاف سيظهر في حال ما اذا تم تغيير قيمة المتغير في داخل A مرات حتى وان تغيرت فيمة المتغير 5 سيتم تنفيذه دائما Next الحلقة، بينما تغيير القيمة يؤثر بشكل آبير على عدد مرات تكرار الحلقة Do... Loop.

تحسين الكفاءة

يطلق على اساليب Optimization بصفة عامة، فان المصطلح تحسين الكفاءة برمجية تتبع لزيادة سرعة تنفيذ الكود او التقليل من استهلاك مصادر النظام وغيرها. اما في موضوع هذه الفقرة فسنناقش تقنيات System Resources من خلال Visual Basic لتحسين الكفاءة والخاصة لعملية الترجمة والتي يوفرها Project الموجودة في صندوق حوار خصائص المشروع Compile خانة التبويب Properties.

Native Codeو P-Code

بتنفيذ Visual Basic - سيقوم مفسر F5 عندما تقوم بتنفيذ البرنامج -بالضغط على بتحويل شيفرة Visual Basic سطر تلو الاخر. قبل عملية تنفيذ السطر، يقوم حتى يفهمها المفسر وينفذ السطر . اما P-Code السطر الى شيفرة من نوع

111

Machine Language فهي تحويل الشيفرة المصدرية الى لغة الالة Native Code هي حجمها P-Code يفهما الجهاز مباشرة . الميزة في الملفات التنفيذية من نوع Visual Basicالصغير نسبيا آذلك توافقيتها المطلقة مع آآواد التنفيذ داخل بيئة . تكون عرضة لاحداث انهيار البرنامج بنسبة اقل بكثير من P-Code المزيد ايضا، آآواد Native Code ابطأ من آآواد P-Code . من ناحية اخرى، آآواد Native Code آآواد Visual Basic حقيقية بل هي لغة مفسر Machine Language لانها ليست آآواد ، فسيوفر لك Native Code فقط. في حالة اختيارك لترجمة الى آآواد من نوع الموجودة في Compile خيارات اضافية تجدها في خانة التبويب Visual Basic Project Propertiesصندوق الحوار :

Optimize for Fast Code:

سيحاول المترجم في هذا الاختيار بتنظيم تعليمات لغة الالة بحيث تعطي اقصى EXEنتائج لسرعة تنفيذ الآآواد بغض النظر عن حجم الملف التنفيذي .

Optimize for Small Code:

سيحاول المترجم في هذا الاختيار بتقليص حجم الملف التنفيذ اقصى ما يستطيع بغض النظر عن سرعة تنفيذ الآآواد فيه.

ملاحظة: توجد علاقة عكسية بين الخيارين السابقين، فغالباً ما يتسبب تقليص حجم الشيفرة في تخفيض سرعة البرنامج، وفي الاتجاة الآخر، غالباً ما يتسبب تحسين سرعة تنفيذ البرنامج إلى زيادة حجم الملف.

No Optimization:

للملف Optimization لن يقوم المترجم باي محاولات لعمليات تح سين الكفاءة التنفيذي.

Favor Pentium Pro:

فهذا الاختيار سيزيد من Pentium Pro اذا أن البرنامج سيعمل على معالج من نوع سرعة تنفيذ التعليمات وخصوصا بغدادية منها.

112

:Create Symbolic Debug Info
سيضيف هذا الاختيار تعليمات اضافية الى الملف التنفيذي لاعطاءه امكانية التنقيح
باستخدام برامج تنقيح الملفات التنفيذية آبرنامج التنقيح الذي توفره بيئة Debug
. نصيحة لك، الغ هذا الاختيار. Microsoft Visual C

إعداداتAdvanced Optimization

من تخصيص بعض خيارات تحسين الكفاءة المتقدمة والتي Visual Basic يمكنك
Advanced Optimizationتجدها في صندوق حوار :

:Assume No Aliasing
هذا الاختيار سيزيد من سرعة تنفيذ البرنامج لكن من الضروري جدا عدم تطبيق
بشكل مبسط - هي عملية استعارة اسم لمتغير عام–مبدأ الاستعارة . والاستعارة
ByRefوارساله الى اجراء بالمرجع :

```
Dim X
Sub MySub )Y As Integer(
Y =4
End Sub
Sub AliasingSub()
`عملية الاستعارة
MySub X
End Sub
```

:Remove Array Bound Checks
مما يزيد من سرعة التعامل Array Index عدم التحقق من رقم فهرس المصفوفة
مع المصفوفات.

:Remove Integer Overflow Checks
عدم التحقق من الق يمة المرسلة الى المتغيرات الصحيحة فيما لو أتت آابر من
المجال لم لا.

:Remove Floating Point Error Checks
Floating Pointمثل الاختيار السابق، لكنه خاص للاعداد من نوع الفاصلة العائمة .
113

:Allow Unrounded Floating Point Operations
للحصول على دقة آابر لاعداد الفواصل العائمة.

:Remove Safe Pentium™ FDIV Checks
سيزيد من سرعة عملية القسمة لكنه قد يؤدي الى نتائج خاطئة لمعالجات
FDIVوالتي تعاني من مشكلة . Pentium
يطلق على اساليب وخوارزميات Optimization اعيد وآرر، مصطلح تحسين الكفاءة
برمجية تتبع لزيادة سرعة تنفيذ الكود او التقليل من استهلاك مصادر النظام
وغيرها. اما في هذه الفقرة فخصصت اساليب لتحسين System Resources
من خلال خانة التبويب Visual Basic الكفاءة والخاصة بعملية الترجمة والتي يوفرها
. تذآر Project Properties الموجودة في صندوق الحوار خصائص المشروع Compile
في احسن الاحوال - نتائج غير متوقعة–ان هذه الاختيار ات قد تسبب مشاآل او
فاحرص على تغييرها بشكل دقيق، ولا تقول ان ترآي ما نبهني!
114

الفصل الرابع

VBAو VBمكتبات

VBA و VB مئات الاجراءات والكائنات المضمنة في مكتبات Visual Basic يوفر لك
والتي لا غنى عنها في برامجك الجدية، صحيح انك تستطيع محاآاة معظم هذه
الدوال بكتابة آواد لانجازها، الا ان استخدام الاجراءات والدوال المشمولة في
يعتبر افضل بكثير من انجازها بنفسك من منظور تحسين Visual Basic مكتبات

، فهذه الدوال صممها مبرمجون محترفون بلغات اخرى مما Optimization الكفاءة
. يأخذك هذا الفصل Visual Basic يجعل تنفيذها اسرع بكثير من آاوادك المكتوبة ب
في جولة مع العشرات من هذه الدوال والاجراءات والتي سأتطرق اليها باختصار
بانتظارك حتى تبحر في MSDN باختصار، اما اذا اردت شرحا وافيا لها فمكتبة
صفحاتها.

التعامل مع الاعداد

والدوال الخاصة بالاعداد Operators عشرات المعاملات Visual Basic يوفر لك
.... الخ، بالاضافة الى دوال رياضية آدوال Byte، Integer، Long باختلاف انواعها
المثلثات او الدوال الاسية.

المعاملات بغدادية

المعاملات الاربع الرئيسية +، -، * و /، وفي حالة تطبيق ها Visual Basic يوفر لك
–على انواع مختلفة من القيم، فان القيم الابسط ستتحول مؤقتا الى الاعقد
، بالنسبة لمعامل القسمة / فهو يقوم Double الى Single و Long الى Integer
، لذلك Double بتحويل جميع القيم المتمثلة في الحدين الايمن والايسر الى النوع
Byte، Integer م ع المتغيرات الصحيحة \ ينصح باستخدام معامل القسمة الصحيحة
فهو اسرع اربع مرات من المعامل /: Long و

115

```
Dim X As Long, Y As Long, Z As Long
Z =X /Y
Z =X \ Y 'هذا اسرع
```

، وفي احيان آثيرة Double فهو يحول جميع القيم الى النوع ^ آذلك معامل الاس
لن تحتاج الا للمتغيرات الصحيحة، لذلك ين صح باستخدام معامل الضرب عوضا عن
الاس:

```
Dim X As Long, Y As Long,
Y =X ^ 3
Y =X *X *X
```

مما لا يعطي دقة Long فيقوم بتحويل القيم الى MOD اما معامل باقي القسمة
، تستطيع تطوير Single و Double في التعامل مع انواع الفاصلة العائمة الاخرى آ
Longدالة اخرى تعود بباقي القسمة للاعداد غير :

```
Function ModEx )dN As Double, dD As Double (As Double
ModEx =dN – Int)dN /dD * (dD
End Function
```

ست معاملات منطقية هي =، Visual Basic بالنسبة للمعاملات المنطقية، فيوفر
. بالنسبة لمعامل المساواة = فهو ليس آمعامل اسناد<> و >، => ،<، =<
Visual Basic القيم الى المتغيرات، فمعامل المساواة = هو المعامل الذي يطبقه
في داخل الجمل الشرطية او حتى اذا سبقه معامل اسناد آخر، فالكود التالي:

```
Dim X As Integer, Y As Integer
X =Y =10
```

يبين لنا ان المعامل = الثاني الموجود في السطر الثاني هو معام ل مقارنة
المساواة وليس اسناد القيم.

ملاحظة: تفرق معظم لغات البرمجة الاخرى بين معامل المساواة ومعامل
معامل المساواة هو == ومعامل C اسناد القيم، فنجد في لغة ال
اسناد القيم هو =.

116

فهي مدعومة ايضا لربط الجمل المنطقية NOT و AND، XOR اما معاملات الربط
التي تمثل قيمة العدد Bits ويمكنك استخدامها للاعداد حيث تؤثر على البتات
Binaryبالنظام الثنائي .

اسبقية المعاملات:

من المفيد ان اذآر هنا، ان من الاخطاء الشائعة التي يقع فيها اغلب المبرمجين ه و
تتم And ، فمقارنة المعامل Or اعلى من المعامل And نسيان ان اسبقية المعامل
هو السابق اي في الجهة Or حتى ولو أن المعامل Or قبل مقارنة المعامل
،ففي هذا المثال: And اليسرى قبل المعامل
Print True Or False And False
بينما النتيجة الحقيقية هي False للوهلة الاولى يعتقد المبرمج ان النتيجة هي
Orيتم اختبار ه قبل المعامل الشرطي . And وذلك، لان المعامل الشرطي True
ولتجنب ذلك، استخدم الاقواس:
Print)True Or False (And False
اي يتم تنفيذه And اعلى من اسبقية المعامل Not المزيد ايضا، اسبقية المعامل
فالعبارة : And دائما قبل معامل
Print Not True And False
False And الاولى فقط حتى تكون True على آلمة Notستقوم بتنفيذ المعامل
Falseوآما هو واضح فنتيجة التعبير هي . And وبعد ذلك يأتي دور المعامل False
ومن ثم عكس النتيجة فتستطيع ان تستخدم And اما اذا اردت تنفيذ المعامل
الاقواس والتي لها الأسبقية الأولى على جميع المعاملات مثل:
Print Not)True And False(
Trueوفي هذه الحالة، سيكون الناتج النهائي هو .
117
الدوال بغدادية
ودالة Abs دالة القيمة المطلقة Visual Basic من الدوال بغدادية التي يوفرها لك
اذا أن 1 فهي تعود بالقيم Sgn ، اما الدالة Exp والدالة الاسية Sqr الجذر التربيعي
اذا أن العدد المرسل لها سالب، وصفر اذا 1العدد المرسل لها موجب، والقيمة -
أن العدد المرسل صفر.
فهي تعود باللوغارثم الطبيعي للعدد، اما للاعداد Log بالنسبة لدالة اللوغارثم
الاخرى، فتستطيع تطوير هذه الدالة:
Function LogEx)dN As Double, dBase As Double (As Double
LogEx = Log)dN / (Log)dBase (
End Function
ايضا، حيث يمكن ان تعود Sqr الكود السابق يذآرني بتطوير دالة الجذر التربيعي
بالجذر النوني للعدد:
Function NthSqr)iNum As Integer, iRoot As Integer (As Double
NthSqr =iNum ^)1 /iRoot(
End Function
' مثال الحصول على الجذر التكعيبي
' 8 لعدد
Print NthSqr)8 , 3(' 2 تعود بالعدد
التي تعود بالقيمة المناسبة استنادا Atn و Sin، Cos، Tan اخيرا الدوال المثلثية
Secالى الزاوية المرسلة لها بالراديان، اما بالنسبة للدوال المثلثية الاخرى ،
، .... الخ فيمكنك اشتقاقها بتطبيقات معادلاته ا المعروفة، هذه واحدة من Cosec
عندي، والبقية عليك:
Function Sec)X As Double (As Double
Sec)X = (1 /Cos)X(
End Function
118
تنسيق الاعداد
التي توفر لك خيارات لا نهائية لتنسيق Format من اقوى دوال التنسيق هي دالة
الاعداد، الحروف، الوقت والتاريخ ايضا، ساسرد لك في ه ذه الفقرة طرق تنسيق
الاعداد فقط.

تتطلب العبارة -او القيمة- و طريقة التنسيق: Format الصيغة المبسطة للدالة
Formatالقيمة) , (طريقة التنسيق
يوجد نوعان من طرق التنسيق . النوع الاول هو التنسيقات القياسية والثاني هو
التنسيقات الخاصة . التنسيقات القياسية عبارة ع ن قيم نحدد نوع تنسيق الارقام آ
لتنسيق الرقم على Currency لتنسيق الرقم بشكل عام او General Number
MSDNشكل عملة وغيرها من القيم التي تجدها في مكتبة :

```
` 1234567
Print Format)1234567, "General Number("
`ر.س.  1,234,567.00
Print Format)1234567, "Currency”(
` 1,234,567
Print Format)1234567, "Standard”(
```

اما التنسيقات الخاصة فهي تنسيقات تحددها بنفسك . والتي تستخدم علامات آ
MSDN... الخ، تجدها ايضا في مكتبة : 0 ، %،#

```
' 1, 234.57
Print Format)1234.567, "#,##.00("
' 23.4%
Print Format)0.234, "#.#("%
' 020.0
Print Format)20, "00#.00("
```

دوال اخرى

، الاولى تحذف الفاصلة وتحول العدد الى Fix و Int من دوال حذف الفواصل الدالتين
عدد صحيح اقل من او يساوي العدد المرسل بينما الثانية تحذف الفاصلة فقط:

119

```
Print Int)1.2 (' 1
Print Int-)1.2 (' -2
Print Fix)1.2 (' 1
Print Fix-)1.2 (' -1
```

التي تمكنك من تحديد عدد VB6 فقد ظهرت في الاصدار Round اما دالة التقريب
الارقام العشرية:

```
Print Round)2.12567, 2 (' 2.13
```

فان الدالتين Octal والثمانية Hexadecimal وعند الحديث عن الاعداد الستعشرية
تحول اعداد النظام العشري الى الانظمة السابقة: Hex و Oct

```
Print Hex$)100 (' 64
Print Oct$)100 (' 144
```

بنفسك: Bin فعليك بكتابة الدالة Binary وللتحويل الى النظام الثنائي

```
Public Function Bin)iNum As Integer (As String
Dim iCounter As Integer
Do
If )iNum And 2 ^ iCounter = (2 ^ iCounter Then
Bin " =1 "& Bin
Else
Bin " =0 "& Bin
End If
iCounter  =iCounter  +1
Loop Until 2 ^ iCounter > iNum
End Function
```

وآبر من او تساوي صفر، 1 فهي تعود بقيمة عشوائية اصغر من Rnd اما الدالة
تستطيع تخصيص مجال معين من الاعداد باستخدام هذه المعادلة:
Int اعلى قيمة)) –+ اصغر قيمة Rnd) * 1(اصغر قيمة +

120
] أكتب شيئا مثل:4 ،2فللحصول على اعداد عشوائية في المجال [-
Print Int)7 *Rnd - +2(
، لكن المفضل استخدام دوال Val اخيرا، دوال تحويل القيم الى اعداد لعل اشهرها
CLng و Integer للاعداد CInt التحويل التي تمكنك من تحد يد نوع القيمة آ
.... الخ. Double للاعداد Long، CDbl للاعداد

التعامل مع الحروف

عشرات الدوال المختصة VBA و VB من منا لا يستخدم الحروف؟ توفر لك مكتبات
. اعرض عليك في هذه الفكرة Strings في التعامل مع المتغيرات والثوابت الحرفية
Find and Replace معظم هذه الدوال بالاضافة الى تطبيق فكرة البحث والاستبدال
ولكني سأبدأ بالمعاملات الحرفية.

المعاملات الحرفية

للقيم الحرفية: Combine Operator يمثل معامل الدمج & الرمز
Dim sMyName As String
"ترآي" = sMyName
"العامري" & sMyName =sMyName
"ترآي العامري" ` Print sMyName
اما معامل الجمع "+" فأنا لا احبذ استخدامه آثيرا، فاذا أن نوع القيم حرفية
"،واذا أتت احدى القيم عددية والثانية حرفية قابلة&فسيتحول الى معامل الجمع "
للتحويل الى عددية فسيكون معامل جمع، اما اذا أتت احدى القيم عددية
لها Type Mismatch والاخرى حرفية لايمكن تحويلها الى عددية، فان رسالة الخطأ
نصيب من الظهور:
Print "20" + "30" ` "2030"
Print "20" + 30 ` 50
Print "X" + 100 `رسالة خطأ

121
، = ... الخ فيمكن تطبيقها على القيم<> ،< بالنسبة للمعاملات المنطقية
او ASCII الحرفية ايضا، حيث تكون قيمة الحروف هي المقابل لها في جدول
:UNICODE
Print "Turki "> "TURKI "` True
Print "Turki "< "TURKI "` False
Print "Turki" = "TURKI "` False
False ` "ترآي" > "احمد"Print
True ` "ترآي" < "احمد"Print
False ` "احمد" = "ترآي" Print
يتجاهل مقارنة شكل الحروف Visual Basic **ملاحظة:** تستطيع ان تجعل
الكبيرة والصغيرة عند استخدام معامل المساواة شريطة كتابة
في منطقة الاعلانات Option Compare Text الكلمة المحجوزة
العامة لكل وحدة برمجية.
احيانا تود تجاهل الدقة التي يفرضها عليك معامل المساواة وتستخدم معامل
أي رقم # الذي يتيح لك استعمال الحروف التعويضية، فيمثل الرمز Like التشابه
والرمز ؟ أي حرف، والرمز * أي عدد معين من الحروف والارقام:
Dim sMyString As String
sMyString =...
A1234" " او " If sMyString Like "A????" Then ... ` "Abcde
Aabce1234" " او " If sMyString Like "A*" Then ... ` "Ab
A1234" " او " If sMyString Like "A####" Then ... ` "A0000
او بامكانك تحديد حروف معينة او مجال معين باستخدام الاقواس [و]:
Dim sMyString As String

sMyString =...
B1234” “ او ” If sMyString Like “]AB[###” Then ... ‘ “A1234
122
BY” “ او ” If sMyString Like “]AB][XY[” Then ... ‘ “AX
D3” “ او ” If sMyString Like “]A-D[#” Then ... ‘ “C9
وحتى يمكنك استثناء حروف معينة او مجال معين باستخدام الرمز !:
Dim sMyString As String
sMyString =...
Z1234” “ او ” If sMyString Like “!]0-9[###” Then ... ‘ “A1234

البحث والاستبدال

InStr تستطيع البحث عن آلمة او حروف معينة داخل قيمة حرفية عن طريق الدالة التي تعود بموقع ذلك الحرف او بداية الكلمة:

Dim lPosition As Long
Dim lStartPoint As Long
lStartPoint =1
"ترآي"), lPosition =InStr)lStartPoint, Text1.Text
If lPosition > 0 Then
Text1.SelStart =lPosition -1
Text1.SelLength =4
End If

فهي شبيهه بالدالة السابقة ولكن عملية البحث تكون InStrRev اما الدالة معآسة -أي تبدأ من نهاية القيمة المرسلة.
التي تمكنك Replace بالنسبة لعملية استبدال النصوص، فلن تجد اسرع من الدالة من استبدال حروف م عينة بحروف اخرى . هنا سنستبدل جميع آلمات "محمد" الى "محمد صلى الله عليه وسلم" الموجودة في أداة النص:
"محمد صلى الله عليه وسلم"), "محمد", Text1.Text =Replace)Text1.Text
123

تنسيق الحروف

ايضا لتنسيق الحروف، ولكن لا توجد بها تنسيقات Format ستستخدم الدالة الذي يمثل حرف او @ قياسية للحروف، اما التنسيقات الخاصة فهي تستخدم الرمز الذي يمثل حرف او لا شئ: & مسافة والرمز

Print Format")ABCD", "@ @ @ @ ("' “A B C D”
Print Format")ABCD", "@ &&& ("' “A BCD”
Print Format")9661234567", "&&&-&-@@@@@@ ("' “966-1-234567”

دوال اخرى

، الحروف اليمنى $ Left من الدوال الحرفية الاخرى دوال استخلاص الحروف اليسرى Right$ و الحروف الوسطى $:Mid

Dim sMyString As String
sMyString =“ABCDEFGHIJKLMNOPQRSTUVWXYZ”
Print Left$)sMyString, 5 (‘ ABCDE
Print Right$)sMyString, 5 (‘ VWXYZ
Print Mid$)sMyString, 20, 5 (‘ TUVWX
Print Mid$)sMyString, 20 (‘ TUVWXYZ

Left و Right وعند الحديث عن حروف لغتنا الجميلة، ف من المعروف ان الدالتين للجملة "مرحب ا" Right $ تعطيان نتائج عكسية مع الحروف العربية . فالاستدعاء سبدأ من الالف فالباء فالحاء الخ .. مما يسبب التشويش على المبرمج العربي . الفكرة بكل بساطة لجعل ها تين الدالت ين تعملان بشكل صحيح مع الحروف العربية، ArLeftو : ArRight هي عن طريق تطوير دالتين عربيتين
Function ArLeft)sString As String, lLength As Long (As String

ArLeft =Right$)sString, lLength(
End Function
Function ArRight)sString As String, lLength As Long (As String
ArRight =Left$)sString, lLength(
End Function

124

Private Sub Form_Click()
' مثال للاستدعاء
Print ArRight $"ترآي العامري"), 4(' ترآي
Print ArLeft $"ترآي العامري"), 7(' العامري
End Sub

ايضا، Statement هي عبارة Mid $ مرة اخرى، فعليك معرفة ان Mid $ اعود للدالة فيمكنك كتابة شيئا مثل:

Dim sMyString As String
sMyString " =abcde"
Mid$)sMyString, 2, 3" = (BCD"
Print sMyString ' "aBCDe"

وبدلا من معرفة ما اذا أن المتغير الح رفي خاليا باستخدام علامات التنصيص، التي تعود بعدد حروف القيمة المرسلة فهي اسرع مرتين من Len استخدم الدالة الطريقة الاولى:

If sMyString ="" Then ...
If Len)sMyString = (0 Then ... 'هذه اسرع

LTrim $ ، المسافات اليسرى RTrim $ اما لحذف المسافات اليمنى استخدم الدالة Trim: $ والمسافات اليمنى واليسرى

sMyString =" 12345 "
Print RTrim$)sMyString (' " 12345"
Print LTrim$)sMyString (' "12345 "
Print Trim$)sMyString (' "12345"

فهي للترميز AscW ، اما ASCII تعود بالعدد المقابل للحرف في جدول Asc الدالة أي العكس:–تعود بالحرف المقابل للعدد ChrW $ و Chr $ ، والدوال UNICODE

Print Asc ("ت") ' 202
Print AscW ("ت") ' 1578
Print Chr$)202 ('ت

125

Print ChrW$)1578 ('ت

اللتان تقومان بتكبير الحروف الانجليزية LCase $ و UCase $ من الدوال الاخرى دالتي وتصغيرها. استخدام هاتان الدالتان مس ألة ضرورية خاصة عند مقارنة القيم الحرفية، فلا تنسى ان "TURKI" : " لا تساوي " Turki"

If Text1.Text ="TURKI" Then ... 'قد لا يكتب المستخدم حروف آبيرة
If UCase$)Text1.Text = ("TURKI" Then ...
If LCase$)Text1.Text = ("turki" Then ...

تقوم بتصغير جميع LCase $ تقوم بتكبير جميع الحروف والدالة UCase $ الدالة الحروف آما رأينا في المثال السابق، اما اذا آنت تريد تكبير الحرف الاول من آل آلمة، فارسل الثابت vbProperCase : الى الدالة StrConv

sMyString " =I like Visual Basic"
Print StrConv)sMyString, vbProperCase (' "I Like Visual Basic"
Print StrConv)sMyString, vbLowerCase (' "i like visual basic"
Print StrConv)sMyString, vbUpperCase (' "I LIKE VISUAL BASIC"

بارسال UNICODE الى ASCII ايضا من تحويل النصوص من StrConv تمكنك الدالة vbFromUnicode بارسال الثابت . ASCII الى UNICODE او من vbUnicode الثابت

ملاحظة: بعض الدوال الحرفية تتوفر منها نسختين، الاولى تعود بقيمة من
. حاول دائما Variant والثانية تعود بقيمة من النوع String النوع
اذا توفرت - عن طريق اضافة–استخدام النسخة الاولى من الدالة
بعد اسم الدالة لترفع عبء تحويل نوع القي مة الذي يقوم به $ الرمز
مما يؤدي الى زيادة سرعة التنفيذ.Visual Basic
تمكنك من فصل جميع الكلمات ونسخ آل آلمة الى مصفوفة، افتراضيا Split الدالة
الفاصل بين الكلمات هي المسافة آما يمكنك تحديد فاصل معين:

```
Dim X ()As String
Dim iCounter As Integer
X  =Split)Text1.Text(
```

126

```
`تحديد فاصل غير المسافة
' X  =Split)Text1.Text, ("*"
For iCounter  =0 To UBound)X(
Print X)iCounter(
Next
```

تعيد الوضع آما أن سابقا: Join واذا ندمت على تقسيم الكلمات، فالدالة

```
sFromArrayToString  =Join )X(
' sFromArrayToString  =Join )X , "*" (`تحديد فاصل غير المسافة
```

، والفرق بينهما هو ان الاولى CStr و Str $ اخيرا، دالتي تحويل القيم الى حرفية
تضيف مسافة يسار العدد المرسل لها ان أن موجب ام الثانية فلا.

التعامل مع الوقت والتاريخ

على قيم الوقت والتاريخ في امكانية تصريح متغير من Visual Basic لا قتصر دعم
، بل يحتوي على عشرات الدوال الخاصة بعرض، تعديل، حساب Date النوع
وتنسيق قيم الوقت والتاريخ.

دوال الوقت والتاريخ

هي قيم تمثل وقت معين او تاريخ معين او آلاهما معا سواء Date القيم من النوع
:# و # او ثابت بين العلامتين Date أتت في متغير من النوع

```
Dim MyDate As Date
MyDate  =#1/20/2001#
Print MyDate
MyDate  =#2:30:00 AM#
Print MyDate
MyDate  =#1/20/2001 2:30:00 AM#
Print MyDate
```

127

وقبل ان ابدأ في الحديث عن دوال الوقت والتاريخ، اود ان اعرفك على الخاصية
والتي تمكنك من تغيير نظام الت اريخ التابع لبرنامجك VBA التابعة للكائن Calendar
الى ميلادي او هجري:

```
Dim MyDate As Date
MyDate  =#1/20/2001 2:30:00 AM#
VBA.Calendar  =vbCalHijri ' هجري
Print MyDate
VBA.Calendar  =vbCalGreg ' ميلادي
Print MyDate
```

مع العلم ان القيمة التي تضعها في هذه الخاصية تؤثر على نوع القيمة التي تعود
بها دوال التاريخ الاخرى ولكنها لا تؤثر في قيم الوقت والتاريخ:

```
Dim MyDate As Date
VBA.Calendar  =vbCalHijri
```

#1= MyDate/ 16/ 1421 # `القيمة هنا بالتاريخ الميلادي وليس الهجري
Print MyDate` مخرجات الامر هنا بالهجري
اللتان تعودان Time و Date والان اعرض لك دوال الوقت والت اريخ مبتدئا بالدالتين
بتاريخ اليوم والوقت الحالي:
Print Date
Print Time
، فهي ستغير قيمة الوقت Statement آعبارات Time و Date اما اذا تعاملت مع
والتاريخ في النظام:
Date =#1/20/2001#
Time =#12:00:00 AM#
تعود بقيمة تشمل تاريخ اليوم والوقت الحالي: Now الدالة
Print Now

128

فهي تعود بعدد الثواني من منتصف الليل حتى وقت استدعائها Timer اما الدالة
أي هي تعمل آعداد، قد تستفيد منها وتطور اجراء انتظار مؤقت قبل تنفيذ آود
معين:

```
Sub Wait )iSeconds As Integer(
Dim sStartTime As Single
sStartTime =Timer
Do :DoEvents :Loop Until Timer – sStartTime > =iSeconds
End Sub
```

ويتم تصفيره من 00:00:00 عبارة عن عداد يبدأ من الساعة Timer تذآر ان الدالة
السابق قد يؤدي الى Wait ، فالاجراء 23:59:59 جديد بعد مرور ثانية من الساعة
قبل نهاية الحلقة الموجودة في الاجراء . Timer حلقة لا نهائية اذا تم تصفير ا لدالة
صحيح ان نسبة حدوث المشكلة السابقة ضئيلة، الا ان تجنب الشوائب امر
ضروري، وآما يقولون "ابعد عن الشوائب وغني لها":

```
Sub Wait )iSeconds As Integer(
Const NUMOFSEC As Single =24 *60 *60!
Dim sStartTime As Single
sStartTime =Timer
Do :DoEvents
Loop Until )Timer +NUMOFSEC – sStartTime (Mod NUMOFSEC > =iSeconds
End Sub
```

مباشرة اذا آنت ترغب في # و # لن تتمكن من استخدام الثابت بين العلامتين
يمكنك DateSerial تعيين قيم لمتغيرات باستخدام التاريخ الهجري، ولكن مع الدالة
VBAالتابعة للكائن : Calendar عمل ذلك فهي تتأثر بقيمة الخاصية

```
Dim MyDate As Date
VBA.Calendar =vbCalHijri
MyDate =DateSerial)1422 , 10 , 27 ( ` 1422 شوال 27
Print MyDate
VBA.Calendar =vbCalGreg
Print MyDate ` 2002 يوافق 11 يناير
```

129

. اما بالنسبة للدالتين TimeSerial آما تتوفر دالة اخرى للوقت هي الدالة
فهما تعودان بقيمة التاريخ او الوقت الموجود في القيمة TimeValue و DateValue
المرسلة اليهما:

```
Print DateValue)Now +2(
Print TimeValue)Now(
```

التي تستخرج جزء معين من قيمة الوقت او DatePart بامكانك استخدام الدالة
فهي Second و Day، Month، Year، Hour، Minute التاريخ، ولكني افضل الدوال

تعود بقيمة اليوم، الشهر، السنة، الساعة، الدقيقة والثانية الموجودة في القيمة المرسلة اليهم:

Print Month)Date(
Print Hour)#1:20:00 AM#(

العمليات بغدادية على الوقت والتاريخ:

آثيرا م ا تحتاج الى اجراء بعض العمليات بغدادية على قيم تاريخية آ الجمع بين تاريخين او طرح تاريخين ، بالاضافة الى مقارنة التواريخ . بامكانك تطبيق ما ذآرته في آما ذآرت - تنقسم–" في الفصل الثاني لان القيمة من هذا النوع Date فقرة "النوع الى قسمين عدد صحيح وعدد عشري ، العدد الصحيح يتعلق بالتاريخ اما العشري فهو خاص بالوقت، جرب استخدام معامل الجمع آما في هذا المثال:

`ساعة من الان 12 اضافة يومين و
Print Now +2 +#12:00#

و DateAdd ولتطبيق عمليات رياضية اآثر دقة وسهولة، يفضل استخدام الدالتين ، الاولى لاضافة تاريخ على تاريخ والثانية لمعرفة الفارق بينهم ا. دالة DateDiff لها صيغة عامة هي: DateAdd

DateAddالفترة) , العدد, (التاريخ

"d يوم " "m" شهر " yyyyالفترة هي الوحدة المستخدم والتي قد تكون سنة " ....الخ، اما العدد فهو عدد الوحدات من الفترة التي تريد اضافته ا، اما التاريخ فهو القيمة الذي تريد اضافة التاريخ عليها:

130

Print DateAdd ")m", 3, Now(

وصيغتها العامة شبيهه DateDiff اما لمعرفة الفرق بين تاريخين فاستخدم دالة بالاولى، الا انها تطلب قيمة التاريخ مكان قيمة العدد:

Print DateDiff")d", #12/ 20/ 2000 #, #2/ 18/ 2001 #)` 60 يوم

تنسيق الوقت والتاريخ

مرة اخرى، وبالنسبة Format لا اعتقد انني بحاجة الى تعريفك على الدالة للتنسيقات القياسية فهي مدعومة لقيم الوقت والتاريخ:

Dim sMyDate As Date
sMyDate =Now
Print Format$)sMyDate, "General Date("
Print Format$)sMyDate, "Long Date("
Print Format$)sMyDate, "Long Time("
Print Format$)sMyDate, "HH:MM -> MMMM DD, YYYY("

FormatDateTime دالة اضافية لتنسيق الوقت والتاريخ هي VB6 آما اضاف الاصدار التي تعود باسم الشهر المقابل للرقم المرسل لها: MonthName والدالة

VBA.Calendar =vbCalHijri
Print MonthName)1 (`محرم
VBA.Calendar =vbCalGreg
Print MonthName)1 (`يناير

التعامل مع الملفات والمجلدات

أي دوال او عبارات جديدة للتعامل مع الملفات والمجلدات، VB6 لم يضف الاصدار Visual فمعظم ما سأسطره في الفقرات التالية توفر منذ الاصدارات القديمة ل . وسأبدأ بعرض دو ال وعبارات تمكنك من التعامل مع الملفات آتعديل Basic خصائصها، اسمائها، الاستعلام عن احجامها .... الخ، ثم اتطرق الى عبارات خاصة

131

وطريقة البحث عن الملفات والمجلدات، ثم اختم الفقرة بطرق Folders بالمجلدات الوصول الى الملفات وتحريرها.

التعامل مع الملفات

من FileCopy من اعادة تسمية الملف او نقله، والامر Name ... As يمكنك الامر
يحذف الملف: Kill نسخ الملف، بينما الامر
`اعادة تسمية ملف
Name "C:\MyFile.EXT" As "C:\MyFile.DAT"
`نقل ملف
Name "C:\MyFile.EXT" As "D:\MyFile.EXT"
`نسخ ملف
FileCopy "C:\MyFile.EXT" As "D:\MyFile.EXT"
`نسخ وتغيير اسم ملف
FileCopy "C:\MyFile.EXT" As "C:\MyFile2.EXT"
`حذف ملف
Kill "C:\MyFile.EXT"
`حذف مجموعة ملفات
Kill ".*TMP"
SetAttr والدالة File Attributes من معرفة خصائص الملف GetAttr تمكنك الدالة
شريطة ان لا يكون مفتوحا:–لتغيير خصائص الملف
Dim sFile As String
sFile ="C:\File.EXT"
If GetAttr)sFile (And vbHidden Then ... `مخفي
If GetAttr)sFile (And vbReadOnly Then ... `للقراءة فقط
If GetAttr)sFile (And vbArchive Then ... `ارشيف
...
SetAttr sFile, vbHidden `مخفي
SetAttr sFile, vbArchive +vbReadOnly `ارشيف وللقراءة فقط
SetAttr sFile, GetAttr)sFile (Xor vbReadOnly `عكس خاصية للقراءة فقط
بوقت وتاريخ FileDateTime بقيمة تمثل حجم الملف، والدالة FileLen تعود الدالة
انشاء الملف:
132
Print FileLen)sFile (
Print FileDateTime)sFile (
هي قابليتها على العودة بحجم الملفات FileLen الميزة التي تع جبني في الدالة
حتى وان أنت مفتوحة، والعيب الذي لا يعجبني في نفس الدالة هو ان القيمة
التي تعود بها هي حجم الملف قبل الفتح -أي لن تعود بقيمة صحيحة في حال
قيام البرنامج بتغيير حجم الملف.

التعامل مع المجلدات

بقيمة حرفية تمثل الدليل الحالي للقرص الذي ينفذ منه CurDir $ تعود الدالة
البرنامج او محرك اقراص آخر ترسله الى الدالة:
Print CurDir $ `الدليل الحالي في القرص الحالي
Print CurDir$)"d" (` Dالدليل الحالي في القرص :
، الاول ChDir و ChDrive وقد ترغب في تغيير الدليل الحالي باستخدام الامرين
لتغيير القرص والثاني لتغيير الدليل:
ChDrive "D:"
ChDir "D:\MsgFolder"
فقط دون ChDir **ملاحظة:** ان قمت بتغيير الدليل الحالي باستخدام الامر
، فستقوم بتغيير الدليل ChDrive تغيير القرص باستخدام الامر
الحالي لذلك القرص فقط، اما الدليل الحا لي الذي ستعود به الدالة
لم يتغير. CurDir$
اذا آنت لا تعرف ما الفائدة من معرفة الدليل الحالي للقرص، فيبدو ان عصر البرمجة
لم تشهده اصابعك الرقيقة . بشكل مبسط، الفائدة التي قد MS-DOS تحت النظام
تجنيها من تغيير الدليل الحالي هو عدم الحاجة الى تحديد مسار الملف ات في ذلك

الدليل، فهذا الكود:

```
ChDrive "C:"
ChDir "C:\UnwantedFolder"
Kill "*.*"
```

133

وهو-سيحذف جميع الملفات الموجودة في الدليل الحالي للقرص
، واتمنى من صميم قلبي الحنون ان لا تجعل دليل النظام C:\UnwantedFolder
هو الدليل الحالي وتطبق الكود السابق. Windows

البحث عن الملفات والمجلدات

من البحث عن الملفات والمجلدات. طريقة استخدامها يتم Dir تمكنك الدالة
بخطوتين: الاولى تحديد الملف /الملفات/المجلد وخصائصها، والثانية باستدعاء الدالة
دون ارسال أي قيمة لها، الكود التالي يبحث عن جميع الملفات التنفيذية الموجودة
C:\WinNTفي المجلد :

```
Dim sFileName As String
'الخطوة الاولى
sFileName =Dir$ )"C:\Winnt\.*EXE"(
'الخطوة الثانية
Do While Len )sFileName(
List1.AddItem sFileName
sFileName =Dir$
Loop
```

تحرير الملفات

بالاضافة اوامر ودوال الاستعلام عن الملفات والمجلدات السابقة، توف ر لك مكتبات
اوامر ودوال اخرى تمكنك من تحرير الملفات لحفظ بيانات برامجك فيها VBA و VB
بالتنسيق والهيئة التي تريده ا. قبل اجراء أي عمليات تحرير على الملف، لابد من
التي ضيغتها: Open فتحه باستخدام العبارة
Open اسم الملف For نوع الوصول Lock الاقفال As #رقم الملف
بالنسبة لرقم الملف، فهو رقم يمثل الملف بحيث يمكنك الوصول اليه من آافة انحاء
البرنامج، ولا يمكن لهذا الرقم ان يشمل آثر من ملف واحد، لذلك حتى تتفادى
التي تعود برقم غير محجوز لفتح FreeFile اخطاء التعارض، يفضل استخدام الدالة
الملف. وبالنسبة للاقفال، فهي تم كنك من تحديد خاصية الاقفال على الملف بحيث
، الكتابة الى Lock Read يمكنك منع آافة البرامج الاخرى من القراءة من الملف

134

. اما نوع Lock Read Write او القراءة والكتابة من والى الملف Lock Write الملف
الوصول، فهي الطريقة التي تود ان تتعامل مع الملف بها وهي ثلاثة انواع:

Sequential Accessالوصول المتتالي :

الاسلوب المتبع مع الوصول المتتالي يعرف بالقراءة والكتابة سطر سطر . ولفتح
للقراءة من الملف، الكلمة المحجوزة Input الملف، استخدم الكلمة المحجوزة
للاضافة الى الملف: Append للكتابة الى الملف والكلمة المحجوزة OutPut

```
Open "MyFile.TXT" For Input As #1
Open "MyFile2.TXT" For OutPut As #2
Open "MyFile3.TXT" For Append As #3
```

- باستخدام Input بامكانك قراءة سطور من الملفات -المفتوحة بالكلمة المحجوزة
EOFحتى نهاية الملف والذي تختبره عن طريق الدالة : Line Input العبارة

```
Dim sLine As String
Open "MyFile.TXT" For Input As #1
Do While Not EOF)1(
Line Input #1, sLine
Text1.Text =Text1.Text & vbNewLine & sLine
```

Loop
الكود السابق لا استخدمه آثيرا فأنا افضل قراءة الملف آاملا بدالة واحدة تسمى
التي تعود بالحجم الكلي للملف: LOF ، واستخدم في ذلك الدالة Input$

```
Dim sFileData As String
Open "C:\MyFile.TXT "For Input As #1
sFileData =Input$ )LOF)1(, 1(
Text1.Text =sFileData
```

و OutPut وبامكانك كتابة سطور الى الملفات -المفتوحة بالكلمة المحجوزة
Print: # - باستخدام العبارة Append

```
Open "C:\MyFile.TXT" For Append As #1
```

135

```
Print #1, Text1.Text
```

التي ستغلق آافة الملفات Close ولا تنسى اغلاق الملف باستخدام العبارة
المفتوحة ان لم ترسل لها رقم ملف معين:

```
` 1 اغلاق الملف رقم
Close #1
`اغلاق آافة الملفات
Close
```

Binary Access **الوصول الثنائي :**

الاسلوب المتبع مع الوصول الثنائي يعرف بالقراءة والكتابة بايت بايت . ولفتح الملف،
للقراءة والكتابة من والى الملف: Binary استخدم الكلمة المحجوزة

```
Open "C:\MyFile.DAT "For Binary As #1
Open "D:\YouFile.DAT "For Binary As #2
```

عملية القراءة والكتابة من الملف متشابهتان من ناحية ا لصيغة الى حد آبير . آل ما
هو مطلوب منك هو معرفة الموقع في الملف وحجم العملية . عندما تقوم بفتح
، وهو اول بايت1 الملف لاول مرة، فان موقع مؤشر القراءة والكتابة من الملف هو
والتي تعود LOF موجود في خارطة الملف . لمعرفة موقع اخر بايت استخدم الدالة
بحجم الملف والذي بديهيا يرمز الى موقع البايت الاخير:

```
Print LOF )1(
Print LOF )2(
```

الذي آنت اقصده من "حجم العملية " هو حجم البايتات التي تريد قراءتها من الملف
للكتابة. راقب Put للقراءة او Get او آتابتها الى الملف . آل هذا يتم باستخدام الامر
هذا الكود:

```
Dim X As Long
Get #1, 1, X
Print X
```

136

```
Get #1, , X ` 5 موقع المؤشر هو
Print X
```

من المهم التنويه هنا بان عملية القراءة من الملف تؤدي الى زيادة الموقع الحالي
للمؤشر بمقدار حجم العملية . ففي السطر الثاني لم احدد موقع المؤشر، لانه
بايت 4بايتات وذلك بسبب ان حجم العملية السابق = 4 سيزيد بشكل تلقائي
. هذا الكود يقوم بقراءة جميع الارقام من ملف وآتابتها Long لقراءة قيمة من نوع
في ملف اخر:

```
Dim lCounter As Long
Dim X As Long
For lCounter =1 To LOF)1(
Get #1, , X
Put #2, , X
Next
```

Seekيمكنك تغيير موقع مؤشر الملف عن طريق العبارة :
Seek #1 , 1 'الى بداية الملف
Seek #2, LOF)2) 'الى نهاية الملف
والتي تتم بنفس الطريقة Strings بدون شك تحتاج الى التعامل مع القيم الحرفية
معلومات عن حجم القيمة الحرفية . يمكنك Visual Basic لكنك بحاجة الى اعطاء
. آما في Fixed Length String عمل ذلك؟ باستخدام النوع الحرفي الثابت الحجم
بايت من الملف: 100 المثال التالي والذي سيقرأ
Dim Y As String *100
Get #1, , Y
واذا آنت لا تفضل استخدام هذا النوع من المتغيرات ، فيمكنك عمل أي شئ تخبر
وذلك عن طريق اسناد أي قيمة 100 ان حجم المتغير الحرفي هو Visual Basic فيه
مؤقتة:
Dim Y As String
137
Y =String)100, ("*"
Get #1, , Y
وعملية كتابة القيم الحرفية الى الملف يمكن لها ان تتم بشكل مباشر مثل:
Put #1 , , "ملف ثنائي"
هو ترميز احادي في الذاآرة مما يؤدي الى بطئ عملية String تذآر دائما ان النوع
التحويل خاصة في حالة آون القيم الحرفية آبير جد ا. ولزيادة السرعة آآثر من
بدلا من Byte ، استخدم عملية المصفوفات للنوع لهذا النوع من القيم -النوع 50%
:String
Dim MyArray)1000 (As Byte
Open "MyFile.DAT "For Binary As #1
' كتابة محتويات المصفوفة الى الملف
Put #1, 1, MyArray
' او قرائة محتويات الملف الى المصفوفة
Get #1, 1, MyArray
Random Accessالوصول العشوائي :
الاسلوب المتبع مع الوصول العشوائي يعرف بالقراءة والكتابة سجل سجل . ولفتح
للقراءة والكتابة من والى الملف مع Random الملف، استخدم الكلمة المحجوزة
ارسال حجم السجل:
Open "C:\MyData.DAT "For Random As #1 Len =200
للقراءة من الملف آما آنت Get للكتابة الى الملف والعبارة Put استخدم العبارة
تفعل مع الملفات الثنائية، ولكن عليك معرفة ان حجم العملية وخطوات انتقال
Lenالمؤشر تتأثر بالحجم المصرح عند فتح الملف باستخدام الكلمة .
138
يفيدك هذا النوع من الملفات لمحاآاة قواعد البيانات بطريقة مبسطة، مثلا يمكنك
والكتابة الى الملف: UDT تعريف ترآيب
Private Type typRCD
sName As String *20
iAge As Integer
End Type
Dim Record As typRCD
Open "C:\MyData.DAT "For Random As #1 Len =Len)Record(
Record.sName = "ترآي"
Record.iAge =99
Put #1, 1, Record
Record.sName = "عبدالله"
Record.iAge =20

```
Put #1, , Record
```

وقراءة السجلات تتم بهذه الطريقة:

```
Dim Record As typRCD
Get #1, 1, Record
Do While Not EOF)1(
Print Record.sName
Print Record.iAge
Get #1, , Record
Loop
```

أسنادات اخرى

مجموعة لا غنى VBA و VB الى جانب الدوال والاجراءات السابقة، توفر لك مكتبات عنها من الكائنات المستخدمة في برامجك الجدية.

139

Appأسناد البرنامج

يمثل البرنامج الحالي الذي يتم تنفيذه . يحتوي على مجموعة App أسناد البرنامج من الخصائص والطرق التي سأتطرق الى بعضها هنا، اما البقية فهي متقدمة بعض الشئ وافضل تأجيلها الى الفصول اللاحقة.

تعود Path ، والخاصية EXE تعود باسم ملف البرنامج التنفيذي EXEName الخاصية بالمسار الكامل الذي نفذ البرنامج منه:

```
Open App.Path & "\" & App.EXEName & ".EXE" For Binary As #1
```

، ففي الكود Path " قبل استخدام الخاصية\من الضروري التحقق من الرمز " وذلك لان مسار البرنامج لن يضاف اليه Path السابق اضفنا هذا الرمز بعد الخاصية هذا الرمز، ولكن تظهر المشكلة في الكود السابق اذا تمت عملية تنفيذ البرنامج " في قيمة الخاصية\للقرص، انظر الى الرمز " Boot Directory من الدليل الجذري اذا نفذ البرنامج من دليل جذري او فرعي: Path

App.Path ="C:\" ' الرمز مضاف

App.Path ="C:\MyProgram" ' الرمز غير مضاف

وحتى تتجنب المشكلة السابقة، طور هذه الدالة وحاول الاعتماد عليه عوضا عن مجردة: Path الخاصية

```
Function PathEx ()As String
If Right)App.Path, 1" = (\ "Then
PathEx =App.Path
Else
PathEx =App.Path & "\"
End If
End Function
Open PathEx & App.EXEName & ".EXE" For Binary As #1
```

تمكنك من معرفة ما اذا أتت نسخة اخرى من البرنامج PrevInstance الخاصية التنفيذي قيد العمل او لا، قد تستطيع منع المستخدم من تشغيل آثر من نسخة للبرنامج في نفس الوقت بهذا الكود:

140

```
If App.PrevInstance Then
MsgBox "لا يمكنك تشغيل نسخة اخرى من البرنامج"
End
End If
```

مع ذلك، لا تثق في الكود السابق آثيرا، فالمستخدم بامك انه تشغيل آثر من نسخة من نفس البرنامج اذا قام بنسخ ملف البرنامج الى مجلد اخر او حتى الى نفس المجلد باسم آخر.

التي تمكنك من TaskVisible من الخصائص التي يمكنك تعديل قيمها الخاصية

- وهي النافذة التي Task List اخفاء او اظهار اسم او رمز البرنامج في قائمة البرامج
تمكنك من عرض جميع البرامج العاملة عن طريق الضغط على المفاتيح
]، فلاخفاء اسم البرنامج في أي وقت: [Ctrl+Shift+ESC] او [Ctrl+Alt+Del
App.TaskVisible =False
Task التي تمكنك من تحديد النص الظاهر في قائمة البرامج Title وآذلك الخاصية
Project ، يكون النص الافتراضي هو النص الموجود عند خانة اسم المشروع List
قبل عملية Project Properties في صندوق حوار خصائص المشروع Name
الترجمة.
هي App من الخصائص الاخرى التي تجدها في صندوق الحوار السابق والكائن
.... الخ، وخصائص حقوق الملكية Major، Minor خصائص رقم اصدار البرنامج
.... الخ وهي للقراءة فقط وقت التنفيذ. LegalCopyRight، Trademarks

Clipboardأسناد الحافظة

بامكانية الاتصال وتبادل البيانات فيما بينها، صحيح Windows تتميز معظم تطبيقات
محدود الامكانيات، الا انه اسلوب Clipboard ان تبادل البيانات عن طريق الحافظة
من نسخ ولصق Visual Basic . يمكنك Windows مازال متبع في معظم تطبيقات
Clipboardالبيانات من والى الحافظة عن طريق الكائن .
SetTextنبدأ أولاً بنسخ النص الى الحافظة باستخدام الطريقة :

```
Clipboard.Clear
Clipboard.SetText Text1.Text, vbCFText
```

141

لمسح محتويات الحافظة قبل Clear **ملاحظة:** ينصح دائما باستخدام الطريقة
نسخ البيانات لها، وذلك لانه في حالات معينة لن تتمكن من نسخ
بيانات جديدة الى الحافظة مالم يتم مسح محتوياتها.
vbCFRTFفاستخدم الثابت : RTF ولنسخ النصوص مع تنسيقها على هيئة

```
Clipboard.Clear
Clipboard.SetText RichTextBox1.Text, vbCFRTF
```

هي المستخدمة: SetData اما لنسخ الصور، فالطريقة

```
Clipboard.Clear
Clipboard.SetData Picture1.Picture
```

هي المستخدمة، ولكن عليك GetText ولاجراء عملية لصق النصوص، فالطريقة
GetFormat اختبار نوع وهيئة البيانات الموجودة في الحافظة باستخدام الطريقة
قبل القيام بعملية اللصق:

```
If Clipboard.GetFormat )vbCFText (Then
Text1.SelText =Clipboard.GetText )vbCFText(
ElseIf Clipboard.GetFormat )vbCFRTF (Then
RichTextBox1.SelRTF =Clipboard.GetText )vbCFRTF(
End If
```

التي تشترط هيئة الصورة: GetData ولصق الصور استخدم الطريقة

```
If Clipboard.GetFormat)vbCFBitmap (Then
Set Picture1.Picture =Clipboard.GetData)vbCFBitmap(
End if
```

يمكنها عرض انواع وهيئات مختلفة من الصور، الا PictureBox رغم ان أداة الصورة
، لذلك vbCFBitmap ان الكود السابق لن يعمل الا اذا أتت هيئة الصورة من النوع
يفضل تمكين جميع الهيئات الاخرى:

```
Dim picFormat As Variant
```

142

```
For Each picFormat In Array)vbCFBitmap, vbCFMetafile, vbCFDIB, vbCFPalette(
If Clipboard.GetFormat)picFormat (Then
Set Picture1.Picture =Clipboard.GetData)picFormat(
```

```
Exit For
End If
Next
```

أسناد الشاشة Screen

أسناد الشاشة يمثل جميع شاشات ونوافذ برنامجك ويحتوي على خصائص تتعلق بالمظهر العام لسطح مكتب نظام التشغيل، آالخاصية FontCount التي تعود بمجموع الخطوط المتوفرة في نظام التشغيل والتي تستخدمها مع الخاصية Font التي تعود باسماء الخطوط:

```
Dim iCounter As Integer
For iCounter =0 To Screen.FontCount -1
List1.AddItem Screen.Fonts)iCounter(
Next
```

الخاصية ActiveForm تمثل نافذة النموذج النشطة في البرنامج، وهي مرجع الى أسناد نافذة النموذج، اما الخاصية ActiveControl فهي تمثل الاداة التي عليها التركيز:

```
Screen.ActiveForm.Caption = "النافذة النشطة"
```

اخيرا، الخاصيتان Width و Height تعودان بارتفاع وعرض الكثافة النقطية للشاشة: Resolution

```
Print "Width " =& ScaleX)Screen.Width, vbTwips, vbPixels(
Print "Height " =& ScaleY)Screen.Height, vbTwips, vbPixels(
```

أسناد الطابعة Printer

الطباعة من المزايا الضرورية التي لابد من توفيرها في برامجك الجدية . بعيدا عن اجراءات API المعقدة، يوفر لك Visual Basic أسناد الطابعة Printer Object والذي من اسمه يعرف غرضه.

143

قبل التوغل في اعضاء أسناد الطابعة اود ان اتطرق الى مجموعة الطابعات Printers Collection . هذه المجموعة تمثل جميع الطابعات الموجودة في الجهاز الحالي . لاتستطيع تعديل مزايا هذه الطابعات بشكل مباشر . فلابد في البداية من تحديد الطابعة وتعيينها للاستخدام ومن ثم تستطيع تعديل الخصائص.

من الخصائص الموجودة في مجموعة الطابعات Printers خاصية اسم الطابعة DeviceName او اسم المشغل DeriverName، رقم المنفذ Port المرآب عليه الطابعة ....الخ:

```
Dim X As Integer
'استخدام المجموعة
' Printers Collection
For X =0 To Printers.Count -1
Print Printers)X.(DeviceName
Next
```

بكل تأكيد تود من المستخدم تحديد ا لطابعة التي يريد استخدامه ا. آل ما عليك هو توفير آود مناسب لتعيين الطابعة الموجودة في المجموعة Printers الى الكائن Printer . هذا مثال لعمل ذلك باستخدام الاداة ListBox:

```
Private Sub Form_Load()
Dim X As Integer
'استخدام المجموعة
' Printers Collection
For X =0 To Printers.Count -1
List1.AddItem Printers)X.(DeviceName
Next
End Sub
```

```
Private Sub List1_Click()
`تحديد الطابعة من المجموعة
Set Printer  =Printers)List1.ListIndex(
End Sub
```

144

لتعديل خصائص ا لطابعة الحالية او Printer والان آل ما عليك هو استخدام الكائن التي تعرف عن ColorMode البدء في عملية الطباعة . من هذه الخصائص ، خاصية والتي PrinterQuality طريقها ما اذا أتت الطابعة داعمة للالوان ام لا . والخاصية تعود بنوع الكثافة النقطية وجودة الطباعة . الخصائص الاخرى تجدها بشكل مفصل MSDNفي مكتبة .

اما عملية الطباعة الفعلية فتتم باستخدام طرق أسناد الطابعة وهي نفس الطرق .... الخ واستخدامها يتم بنفس Print، Line، Cirlce الموجودة في أسناد النموذج الطريقة التي استخدمناها مع نافذة النموذج في الفصل الثاني "النماذج والادوات ". لبدء EndDoc بعد ان تنتهي من ارسال البيانات الى الطب اعة، استخدم الطريقة عملية الطباعة الفعلية:

```
Printer.RightToLeft =True
Printer.FontSize  =40
Printer.Print "ترآي العامري"
`ابدأ عملية الطباعة
Print.EndDoc
```

واضح من اسمها NewPage تقوم بانهاء عملية الطباعة، والطريقة KillDoc الطريقة انها تخرج صفحة جديدة.

اآتشاف الاخطاء

مما لا شك فيه، ان من اهم اسباب انتشار الشعيرات البيضاء في رؤوس المبرمجين هي الاخطاء البرمجية . فكم من مبرمج آسر شاشة جهازه بسبب آثرة الاخطاء غير المتدارآة في برنامجه، وآم من مبرمج توقف عن إآمال مشاريعه بسبب عدم معرفة مصدر الخط أ. كتابة برنامج دون اخطاء شيء يتحقق في الخيال فقط! لكن آلما زادت احتياطاتك لتفادي الاخطاء قلت نسبة ظهورها.

فكرة عامة

تصنف الاخطاء في أي لغة برمجة الى صنفين على اساس وقت حدوثها اما في وقت التصميم او وقت التنفيذ . هذه الاخطاء تسبب انهيار برنامجك وانهاء تنفيذه . بالاضافة الى ذلك، يوجد نوع من الاخطاء التي لا تظهر لك بشكل مباشر تعرف . لنلقي نظرة على هذه الاصناف: Bugs بالشوائب

145

Design Time Errors اخطاء وقت التصميم :

وهي اسهل انواع الاخطاء آتشافا Syntax Errors وتعرف ايضا بالاخطاء النحوية واصلاحا. وقت حدوث هذه الاخطاء يكو ن في مرحلة التصميم او الترجمة للبرنامج . سببها الرئيسي في طريقة كتابة العبارات البرمجية الخاطئة . فمثلا قد تكتب اسم Nextبدون اقفالها باستخدام . For دالة ليست موجودة، او تنشئ حلقة تقنية في قمة الروعة هدفها قنص Visual Basic توفر لك بيئة التطوير المتكاملة ل ENTERهذه الاخطاء تلقائيا بمجرد الوقوع فيها وذلك بعد الضغط على المفتاح [ ]. [ستلاحظ ظهور رسالة توضح لك ENTERواضغط مفتاح [4== X مثلا، قم بكتابة الخطأ وقد قلب لون السطر بالاحمر . تعرف هذه التقنية بالتدقيق النحوي التلقائي Auto Syntax والتي تستطيع الغ ائه ا ع ن طريق الاختيار Auto Syntax Check . لا اعتقد انك Options في نافذة الخيارات Editor الموجود في خانة التبويب Check ستلغيها يوما من الايام اليس آذلك؟!

Run Time Errors اخطاء وقت التنفيذ :

وقت ظهور هذه الاخطاء مختلف . فلن تظهر الرسالة المزعجة السابقة وقت كتابة

الكود و انما في وقت التنفيذ . عندما يصل المفسر عند سطر صحيح نحويا لكنه
ويظهر تحديد لمكان Run Time Error خاطئ منطقيا ستظهر رسالة خط أ بعنوان
السطر الذي وقع فيه الخطأ. مثلا اأكتب هذا الكود:

```
Dim X As Byte
X =256
```

من الواضح ان الصيغة النحوية لهذا الكود صحيحة لكن من الن احية ا لمنطقي ة خط أ.
وذلك لان القيمة OverFlow جرب تنفيذ البرنامج وستلاحظ ظهور رسالة خطأ
. طبعا اخطاء وقت 255 هي Byte القصوى التي يمكن ان يحملها أي متغير من نوع
التنفيذ آثيرة جدا جدا، فانت عندما تصمم البرنامج تتوقع ان آل الاحتمالات
الخارجية آما هي في حالة تصمي م البرنامج . مثلا لو وجد في احد سطور برنامج
امر يقوم بمسح ملف معين وأكتبت هذا الكود:

```
Kill "FileName.EXT "
```

قد عمل معك بالشكل المطلوب، لكن افترض ان الملف لم يكن موجود؟ فان رسالة
الخطأ سيكون لها نصيب من عمر تنفيذ البرنامج . فلو آنت ذآيا ستتأآد من وجود
ومن ثم حذفه: Dir الملف باستخدام دالة

146

```
If Dir$)"FileName.EXT" (Then Kill "FileName.EXT "
```

يبدو ان ذآائك خارق جدا يا قارئ هذه السطور لكن مهلا آأكتب هذه السطور لديه
مدعومة به ReadOnly شئ اخر ليخبرك به . ماذا لو أن الملف موجود لكن خاصية
أي انه غير قابل للحذف؟؟ ارأيت رس الة الخطأ ستظهر من جديد . اذن ستستخدم
ReadOnlyذآائك الخارق وتتأآد من وجود الملف ومن ثم من عدم وجود خاصية :

```
If Dir$)"FileName.EXT" (Then
If Not )GetAttr")FileName.EXT" (And vbReadOnly (Then
Kill "FileName.EXT"
End If
End If
```

حسنا يا قارئي الفاض ل، لك مني فائق التقدير والاحترام على محاولتك الرائعة
لتجنب الخطأ لكن مع ذلك فهناك احتمال اخر لوقوع الخط أ! افترض ان الملف
وأتت عليه خاصية Process يتم استخدامه من قبل برنامج اخر FileName.EXT
الاقفال -أي مسموح للبرنامج الذي يستخدمه فقط - فانك لن تستطيع فتح ا لملف
وستظهر رسالة الخطأ التي اخبرتك بها واآون قد غلبتك في هذا التحدي.
القضية ليست من يغلب من، فكلنا مبرمجين ننسى آثير ا. لكن لابد لأي مبرمج من
وضع جميع وآافة الاحتمالات الممكنة لتفادي وتجنب الاخطاء آما سنرى لاحقا.

Bugsالشوائب :

قد يكون الكود سليم من ناحية نحوية ولا توجد به أي اخطاء في وقت التنفيذ لكن
به شوائب . لا يوجد برنامج الا وبه شوائب . الشوائب هي اخطاء في سلوك تنفيذ
البرنامج لكنها لا تسبب في ايقافه، وهي صعبة الايجاد والاآتشاف . لذلك، تجد
توزع على اش خاص Beta غالبا في البرامج التجارية الكبيرة صدور نسخ تجريبية
وشرآات معينة الهدف منها تجربة البرنامج والتحقق من واآتشاف الشوائب
الموجودة فيه . من آبر الاخطاء الذي يقع فيها المبرمج هي محاولة آتشاف
الشوائب بنفسه ، لأنك لن تستطيع اآتشاف الشوائب الا عن طريق غيرك ، ففي
حالة تجربة برامجك الشخصية فانك اآثر من يعرف طريقة التعامل معها، لكن في
حالة وجود نسخة من البرنامج عند شخص اخر فالوضع يختلف، وتبدأ الشوائب
بالظهور لديه.
به شوائب. هناك الكثير من الشوائب التي تكتشف شهريا وتصدر Visual Basic
، بعضها تم MSDN تقارير عنها تجدها بشكل دوري في مكتبة Microsoft شرآة

147

اصلاحه و ال بعض الاخر لا. المقصد من هذا الكلام، انه مهما أن مستواك في
البرمجة لابد من وجود شوائب في برنامجك.

يوجد الكثير من الكتب التي تناقش مسألة الشوائب البرمجية وآيفية تفاديها -اقصد الاقلال منها- الا انها مواضيع خارج نطاق الكتاب.

Errالكائن

FileName.EXTعودا الى موضوع اخطاء وقت ا لتشغيل وبالتحديد في مثال الملف ، بدلا من كتابة عشرات الاسطر للتأآد من قابلية حذف الملف، استخدم أسناد الخطأ . قبل تطبيق هذا الكائن ، عليك معرفة أن آل خطأ من اخطاء وقت التشغيل له Err رقم خاص يميزة عن غيرة من الاخطاء به وآذلك وصف نصي مختصر للخط أ، وعند . عند رغبتك Err حدوث الخطأ يتم وضع هذه البيانات -الخاصة بالخط أ- في الكائن في الاستمرار في عملية تنفيذ البرنامج حتى عند وقوع الخطأ لابد من كتابة عند بداية آل اجراء تتوقع حدوث خطأ فيه حتى On Error Resume Next التعليمة يستمر في تنفيذ سطور البرنامج راقب هذا المثال:

```
On Error Resume Next
Kill "FileName.EXT"
If Err Then
MsgBox Err.Description
Err.Clear
End If
```

Err هنا سنقوم بمحاولة حذف الملف ، ان لم يستطع البرنامج فعل ذلك فان الكائن سيحتوي على خصائص تتعلق بذلك الخط أ و سنظهر رسالة توضح وصف الخط أ. من حتى نخبر Clear عن طريق استدعاء الطريقة Err المهم التأآد من تنظيف الكائن البرنامج اننا انتهينا من قنص الخطأ وانه لا يوجد خطأ اخر.

If Err Then اما اذا أتت آلواد الاجراء طويلة ولا تود ان تكتب الجملة الشرطية والتي تؤدي الى الانتقال الى On Error Goto X مرات متعددة، فيفضل استخدام سطر معين في حال حدوث أي خطأ في تنفيذ آلواد الاجراء:

148

```
Sub LongSub ()
On Error Goto X:
...
...
...
X:
MsgBox Err.Description
Err.Clear
End Sub
```

149

الفصل الخامس

OOPالبرمجة أسنادية التوجه

هي لغة برمجة مبنية على Visual Basic ان VB3 و VB1، VB2 عرّفت الاصدارات ، اما نقطة التحول OBP تختصر –Object Based Programming Language الكائنات على انها لغة Visual Basic والذي مكننا من اعتبار VB4 أتت منذ انطلاق الاصدار OOPتختصر –- Object Oriented Programming Language برمجة أسنادية التوجه على هذه اللغة، الا ان البعض يعترض على Classes بعد اضافة ميزة تعريف الف ئات OOP لعدم دعمها لبعض الميزات الاساسية للغات OOP بانها Visual Basic وصف .... الخ، من ناحية اخرى فهو OverLoading ، إعادة التعريف Inheritance آالوراثة عن ط ريق تعريف Encapsulation وهو التغليف OOP يدعم المبدأ الاساسي للغات Interfacesوالواجهات . Classes الفئات

Cحقيقية آلغات ++ ، OOP لا يعتبر لغة أسنادية التوجه Visual Basic خلاصة القول،

وتطبيق معظم مبادئه ا. OOP ، ولكنه يمكنك من محاآاة لغات JAVA او SmallTalk
للجميع - نحو برمجة أسنادية التوجه "، Visual Basicومن منطلق عنوان هذا الكتاب "
Visual وتطبيقها ب OOP فهذا الفصل هو مدخلك الرئيس الى البرمجة أسنادة التوجه
، وستكون جميع الفصول اللاحقة من هذا الكتاب مبنية على هذا الفصل. Basic
نظراً لأن جميع الفصول اللاحقة من هذا الكتاب ستكون مبنية على ما تعلمته من
هذا الفصل، فاني ارجو منك أن تتقبل مني هاتين النصيحتين:
- لا تحاول تجاوز إي فقرة . اذا شعرت أنك غير مستوعب للفكرة، حاول قراءة
الفقرة جديد مع تطبيق الأمثلة المدرجة.
- حاول ابتكار امثلة جديدة من وحي افكارك، وقم بتطبيقها، لتتمرس على هذا
الأسلوب من البرمجة.

مقدمة الىOOP

فتستطيع الانتقال OOP اذا آنت على دراية آ افية بمصطلح البرمجة أسنادية التوجه
فيمكنني ان اعرف OOP الى فقرة بناء اول فئة مبسطة، اما ان آنت جديدا على
لك البرمجة أسنادية التوجه على انها برمجة موجهه نحو أسنادات او اهداف، فكل شئ

150

له بيانات وافعال خاصة به أي Thing او شئ Object في برنامجك عبارة عن أسناد
اشبه بالعالم الحقيقي الذي تراه يوميا، فالانسان أسناد له صفات معينة (خصائص
) آالاسم، العمر، اللون، الطول، الوزن، .... الخ، وله افعال يقوم بها Properties
) آالمشي، الكتابة، الضحك، البكاء، النوم، .... الخ، آما ان الانسان Methods (طرق
) تؤثر فيه وينتج عنها ردود فعل آاستقبال رسالة Events تحدث عليه وق ائع (احداث
مفرحة او محزنة، التعرض لجلطة في المخ، وصول لكمة خطافية في الفك الايمن،
صفعة قوية في الخد الايسر، .... الخ.
، فهي تحتوي على خصائص تحوي بيانات Visual Basic آذلك الحال مع أسنادات
.... الخ، وطرق لتفعل افعال خاصة بها Left، Height، BackColorخاصة بها مثل :
Click، MouseMove.... الخ، واحداث تقع عليها آ ، Move، Refresh، ZOrderمثل:
، .... الخ تنتج عنها ردود فعل خاصة. KeyPress

لماذاOOP ؟

آثيرة جدا ولكني ساختصر ثلاثة منها: OOP بصراحة الفوائد التي تجنيها من
فالبرنامج OOP - عندما تكبر حجم البرامج تزداد عملية ادارتها تعقيدا، لكن مع
يتكون من مجموعة أسنادات بحيث انه لو حدثت مشكلة في احدها فكل ماهو
مطلوب هو تعديل ذلك الكائن دون ان تتأثر الكائنات الاخرى، وحتى لو اردت
تطوير احد الكائنات فلست مضطرا الى تنقيح آلاف الاسطر من البرنامج، وآل ما
يتوجب عليك القيام به هو الانتقال الى آود الفئة وتطويره فقط.
- تصميم البرامج والتخطيط لبنيتها اصبحت اسهل من البرمجة الاجرائية واقرب
الى العالم الحقيقي، فعندما تخطط لبرنامج جديد فنظرتك ستكون بعيدة عن
الاآواد وقريبة الى التصميم بحيث تنجز مه امك بسرعة آابر وسهولة آاثر .
فعندما تصمم فئة جديدة، فلن يشغلك أي آود او متغير خارج هذه الفئة قد يؤثر
على سلوك تنفيذ الاآواد، وسيكون آل ترآيزك على هذه الفئة وأنها الجزء
وعدم تأثر Bugs الوحيد الموجود في البرنامج، مما يقلل نسبة الشوائب
متغيرات وبيانات برنامجك.
دون أسنادات واردت تغيير Visual Basic - ستجعل حياتك اسهل، فلو تخيلت OOP
اسم النافذة، فقد تكتب شيئا مثل:

151

```
Dim hWND As Long
hWND  =FindWindow )"Form1"(
ChangeCaption  )hWND, "Main Menu"(
```

Caption وتقوم بتغيير خاصيته Form1 فانك تتحدث عن أسناد اسمه OOP لكن مع

الى الاسم الذي تريده بسهولة شديدة.

OOPسمات

بشكل–، فالفئة Object والكائن Class من الضروري ان اوضح الفرق بين الفئة
مبسط- هي مجرد وصف لخصائص، طرق واحداث الكائن، بينما الكائن هو وحدة
تحتوي على بيانات واآواد معرفة في الفئة . اعود للمثال السابق، فالانسان هو فئة
واعوذ–خلقها الله عز وجل واصفة لخصائص، طرق واحداث أسنادات مشتقة منها، فأنا
.... 99 بالله من آلمة ان ا- أسناد لدي خصائص من فئة الانسان آالاسم ترآي، العمر
الخ، وانت ايضا أسناد لديك خصائص من نفس الفئة "الانسان " آاسمك س، عمرك
هي Text2 و Text1 ، فادوات النص Visual Basic ص .... الخ. آذلك الحال مع
هي أسنادات Labe3 و Label1، Label2 ، وادوات العنوان TextBox أسنادات من الفئة
Labelمن الفئة .
OOPبودي ان اعرض عليك باختصار السمات الثلاث ل :

التغليف:

Putting بوضع جميع الاشياء معا OOP في لغات Encapsulation يقصد بالتغليف
، بحيث تحقق استقلالية الكائن المطلقة ببياناته الخاصة به everything together
وحتى اآواده، من المزايا التي يقدمها لك التغليف هو امكانية تطوير البنية التحتية
للكائن بدون ان يتأثر ترآيب برنامجك ودون الحاجة الى تعديل سطر واحد من اآواد
البرنامج، مثلا لو قمت بتصميم فئة لل بحث عن الملفات واعتمدت عليه بدرجة آبيرة
في برنامجك، وبعد فترة من الاختبارات والتجارب القوية لاحظت بطء في عملية
التنفيذ، فكل ما ستفعله هو تعديل البنية التحتية للفئة الخاصة بالبحث وتطوير
خوارزميات اآوادها دون تغيير سطر واحد من سطور البرنامج الاخرى والتي
تستعمل هذه الفئة بالتحديد.
آلما زادت استقلالية الفئة، آلما زادت آفاءة اعادة استخدامها في برنامج آخر
. مبدأ اعادة استخدام Code Reusability وتطبيق اسلوب اعادة استخدام الآواد
الآواد من احد المبادئ الضرورية التي يتوجب عليك محاولة والتعود على تطبيقها
دائما في برام جك ومشاريعك اليومية، بحيث تتمكن من الاستفادة من الفئة التي
صممتها في آثر من مشروع وآثر من برنامج . وحتى تنشئ فئة قابلة لاعادة
الاستخدام، حاول دائما وقبل ان تبدأ بكتابة سطر واحد من الفئة باخذ احتياطاتك

152

للمستقبل واسأل نفسك اسئلة شبيهه ب : آيف يمكنني الاستفادة من هذه الفئة
في برنامج آخر؟ آيف اسمي واحدد الخصائص، الطرق والاحداث بحيث تكون قابلة
للعمل مع آثر من برنامج وقابلة للتطوير ايضا ؟ آيف اجعل هذه الفئة مستقلة قدر
المستطاع عن أي اآواد او أسنادات اخرى في البرنامج بحيث يمكنني استخدامها
في برنامج آخر؟ ....الخ م ن الاسئلة والاعتبارات التي لابد من وضعها في الاعتبار
قبل بناء الفئة وعند كتابة آل اجراء من اجراءاتها.

تعدد الواجهات:

هو قدرة الفئة على احتوائها آثر من Polymorphism بواسطة مبدأ تعدد الواجهات
واجهة بحيث تمكنك من توحيد عدة فئات مختلفة باسماء اعضاء متشابهه، فلو
ستجد انها مختلفة المهام والانجازات الا Visual Basic امعنت النظر قليلاً في ادوات
مما Click و Left، Move انها تحتوي على خصائص، طرق واحداث مشترآة آ
يسهل عليك آمبرمج حفظها وتوحيد الاجراءات التي تستخدم هذه الاعضاء . الفصل
القادم يناقش مبدأ تعدد الواجهات بالتفصيل.

الوراثة:

هي قدرة فئة على اشتقاق اعضاء من فئة ام بحيث تزيد من Inheritance الوراثة
قوة الفئة الوارثة وتضيف اعضاء جديدة للفئة الام، فلو أن لديك فئة قوية واردت
اضافة طريقة او خاصية لها، فلا يوجد داعي لاعادة بناء الفئة من جديد واضافة
الخاصية او الطريقة المطلوبة، فكل ما ستقوم به هي عملية انشاء فئة خالية

تضيف اليها الخاصية او الطريقة التي تريدها ومن ثم تشتقها من الفئة التي تريد تطويرها واضافة الخاصية او الطريقة لها. الفصل القادم يناقش مبدأ الوراثة بالتفصيل.

بناء اول فئة مبسطة

آي نبدأ بتصميم اول فئة تمثل شخص سنسميها Visual Basic والان شغل
، ومن صندوق الحوار Project من قائمة Add Class Module . اختر الامر CPerson
لتظهر لك Open وانقر على الزر Class Module قد- يظهر امامك، اختر الرمز–الذي
] لعرض نافذة خصائص الفئة، F4نافذة آواد تعريف الفئة، اضغط على المفتاح [
، واأكتب هذا الكود في الفئة: CPerson الى Class1 وعدل خاصية الاسم من

```
Public sName As String
Public dBirthDate As Date
```

153

و sName تحتوي على الخاصيتين CPerson وبهذا نكون قد انجزنا اول فئة بالاسم
في أي مكان داخل مشروعك، CPerson . تستطيع استخدام الفئة dBirthDate
التابع لنافذة النموذج واأكتب هذا الكود: Click اذهب الى الحدث

```
Private Sub Form_Click()
Dim Turki As New cPerson
Dim Khaled As New cPerson
= "ترآي العامري"Turki.sName
Turki.dBirthDate  =#1/1/1900#
= "خالد الابراهيم"Khaled.sName
Khaled.dBirthDate  =#1/1/1979#
Print Turki.sName, Turki.dBirthDate
Print Khaled.sName, Khaled.dBirthDate
End Sub
```

من الفئة التي صممناها Khaled و Turki قمنا -في الكود السابق - بانشاء أسنادين
لكل dBirthDate و sName ، ومن ثم قمنا بتعيين قيم للخاصيتين CPerson للتو
و Turki أسناد على حدة، وختمنا الكود بطباعة قيم الخصائص التابعة للكائنين
.Khaled

بشكل جاد - لطباعة قيم–صحيح ان الفئة السابقة لن تطبقها في حياتك البرمجية
متغيرات، الا ان الغرض الاساسي هو مجرد توضيح فكرة الفئات وطريقة استخدامها.

بناء الفئات

سنبدأ بالتوغل في Classes والان بعد ان عرفتك على الفكرة الاساسية من الفئات
تفاصيل بناء خصائصها، طرقها واحداثها حتى تزيد من قوة الفئة.

بناء الخصائص

السابق، فسنلاحظ ان المبرمج يستطيع اسناد CPerson اذا عدنا الى مثال الفئة
، وقد يعطي فرصة للمستخدم بادخال العمر من dBirthDate أي قيمة للخاصية
خانة نص:

```
Turki.iAge  =CDate  )Text1.Text (
```

154

المشكلة في الكود السابق، ان المستخدم بامكانه ادخال أي عدد يمثل تاريخ
ميلاد الشخص وقد يكون تاريخ لم يحل بعد، لذلك عليك التحقق من تاريخ الميلاد
dBirthDateفي آل مرة تمكن المستخدم من ادخال قيمة للخاصية :

```
If CDate )Text1.Text  (> Date Then
"خطأ في القيمة" MsgBox
Else
Turki .dBirthDate =CDate)Text1.Text(
End If
```

يعيب الكود السابق انه يلزمك بعملية التحقق من القيمة في آل مرة تريد اسناد

، والحل هو باستخدام نوع خاص من الاجراءات يس مى dBirthDate قيمة للخاصية الخصائص والتي تمثل خصائص الكائن لتحميه من القيم الخاطئة:

```
Private m_dBirthDate As Date
Public Property Get dBirthDate ()As Date
dBirthDate =m_dBirthDate
End Property
Public Property Let dBirthDate )ByVal dNewValue As Date(
If dNewValue > Date Then
MsgBox "خطأ في القيمة"
m_dBirthDate =Date
Else
m_dBirthDate =dNewValue
End If
End Property
```

آما في الكود–من داخل الفئة MsgBox **ملاحظة:** في الحقيقة، اظهار رسالة السابق- يعتبر اسلوب غير احترافي وتصميم سيء جدا للفئات خاصة عندما تزيد احجامها، ويفضل ارسال رسالة خطأ بالطريقة MsgBox . الا انني استخدمت الدالة MsgBox بدلا من Err.Raise في المثال لتقريب الفكرة اليك.

155

سيناريو تنفيذ الاجراءات السابقة سيكون آالتالي : في آل مرة تقوم بتعيين او Let dBirthDate ، سيتم استدعاء الاجراء dBirthDate اسناد قيمة جديدة للخاصية ، وفي أول مرة تقوم بقراءة قيمة dNewValue وارسال القيمة الجديدة الى المتغير والذي يعود بقيمة Get dBirthDate ، سيتم استدعاء الاجراء dBirthDate الخاصية دون الحاجة لتعريف متغير خاص iAge الخاصية. بامكانك ايضا اضافة خاصية جديدة لها: Private

```
Public Property Get iAge ()As Integer
iAge =DateDiff")yyyy", m_dBirthDate, Date(
End Property
```

Let iAge وتجاهلت الاجراء Get iAge تلاحظ انني لم استخدم الا اجراء واحد وهو ، فلو حاول المبرمج Read Only للقراءة فقط iAge وذلك لانى اريد ان اجعل الخاصية Read Only Propertyتعيين او كتابة قيمة جديدة للخاصية ستظهر رسالة خطأ :

```
Print Turki.iAge 'ممكن جدا
Turki.iAge =80 'رسالة خطأ
```

Property دون الاجراء Property Let وبامكانك تطبيق العكس، أي استخدام الاجراء Write Only Propertyلتجعل الخاصية للكتابة فقط : Get

```
Private m_sPassword As String
Public Property Let sPassword )sNewValue As String(
m_sPassword =sNewValue
End Property
```

Functions او Sub's المزيد ايضا، يمكنك التعامل مع اجراءات الخصائص آاجراءات Parametersعادية لتمكنها من استقبال قيم :

```
Private m_sAddress )2 (As String
Public Property Get sAddress )iIndex As Integer (As String
sAddress =m_sAddress )iIndex (
End Property
```

156

```
Public Property Let sAddress)iIndex As Integer, sNewValue As String(
m_sAddress )iIndex = (sNewValue
End Property
```

بهذه الطريقة: sAddress وبامكانك استدعاء الخاصية
Turki.sAddress)0 = (" شارع الحقيقة–حي الوهم "
Turki.sAddress)1 = (" باآستان–ولاية فلوريدا "
Turki.sAddress)2 = (" 999 هاتف منزل"
وعند الحديث عن الخصائص التي تمكنك من اسناد قيم لكائنات، فعليك استخدام
Property Letعوضا عن الاجراء : Property Set الاجراء

```
Private m_PersonParent As CPerson
Public Property Get PersonParent ( )As CPerson
Set PersonParent  =m_PersonParent
End Property
Public Property Set PersonParent )ByVal objNewValue As CPerson (
Set m_PersonParent  =objNewValue
End Property
```

Property Attributesمواصفات الخصائص :

Procedure Attributes بامكانك تعديل مواصفات الخصائص عن طريق صندوق الحوار
بعد تحريك مؤشر الكتابة الى مكان اجراء Tools والذي تصل اليه من القائمة
، اخفاء Default Propertey الخاصية. من هذه المواصفات: جعل الخاصية اف تراضية
، كتابة وصف للخاصية Object Browser الخاصية من نافذة مستعرض الكائنات
....الخ، بامكانك استكشاف باقي الخيارات في صندوق الحوار، وعليك ان تعلم علم
. فقط ولن تظهر لك في CLS اليقين ان جميع هذه الخيارات ستحفظ في ملف الفئة
نافذة محرر آواد الفئة، فلو قمت بعملية نسخ ولصق آواد الفئة الى فئة اخرى،
عليك اعادة عملية تحرير مواصفات الفئة.

157

بناء الطرق

معرفة داخل الفئة، ولا Function او دوال Sub's ماهي الا اجراءات Methods الطرق
" لتفاصيل بناء BASIC اعتقد انك بحاجة الى اعادة الفصل الثالث "لغة البرمجة
CPersonالاجراءات والدوال. مع ذلك، هذا مثال لطريقة تابعة للفئة :

```
Public Sub SetData)sName As String, dBirthDate As Date, sAddress As Variant(
Me.sName  =sName
Me.dBirthDate  =dBirthDate
Me.sAddress)0 = (sAddress)0(
Me.sAddress)1 = (sAddress)1(
Me.sAddress)2 = (sAddress)2(
End Sub
```

بامكانك استدعاء هذه الطريقة بدلا من تعيين آل خاصية على حده:

```
`بدلا من تعيين الخصائص
Turki.sName  =txtName.Text
Turki.dBirthDate  =CDate  )txtBirthDate.Text (
Turki.sAddress )0 = (txtAddress1.Text
Turki.sAddress )1 = (txtAddress2.Text
Turki.sAddress )2 = (txtAddress3.Text
`استدعي الطريقة
Turki.SetData txtName, CDate)txtBirthDate(, Array)txtAddress1, _
txtAddress2, txtAddress3(
```

ملاحظة: حتى لو لم تقتع بفكرة تعيين الخصائص باستخدام الطرق آما في
المثال السابق، تذآر ان استدعاء الطريقة السابقة اسرع بخمس
مرات من تعيين قيمة آل خاصية على حده، وستؤثر هذه السرعة
DCOMاو . COM آمكونات–آلما أتت آواد الكائن ابعد

158

بناء الاحداث

حدث الانشاء VB4 عندما نربط بين آلمتي الاحداث والفئات يتبادر لذهن مبرمجي
. لكن مع الاصدارات الاحدث، اصبحت Class_Terminates والانهاء Class_Initialize
Clients الفئات قابلة على انشاء وتعريف احداث جديدة قابلة للتصريح من العملاء
والذين قد تكون انت احدهم.–المستخدمين لتلك الفئة
الفكرة ليست صعبة او مختلفة عن الاحداث الموجودة في الادوات ، لن أخذ مثلا نافذة
ولاشتقاق واستخدام ذلك الحدث Click ، تم تعريف حدث فيها باسم Form النموذج
آل ما هو مطلوب منك وضع اسم الكائن ثم شرطة سفلية ومن ثم اسم الحدث
آما في هذا الكود:

```
Private Sub Form_Click()
`اشتقاق حدث النقر من أسناد النموذج
End Sub
```

قد يحتوي على Sub من هنا يتضح لنا ان الحدث بكل بساطة عبارة عن اجراء
lKeyPressو . MouseDown آالموجودة في حدث Parameters متغيرات اضافية
موجود في DataHasBeenSent الذي سنفعله هنا بالضبط هو تعريف حدث باسم
. ولتعريف هذا ا لحدث الجديد في الفئة سنستخدم الكلمة المحج وزة CPerson الفئة
:Event

```
`تعريف حدث جديد
Event DataHasBeenSent)objTo As CPerson, bSuccess As Boolean(
```

لكن مهلا ! متى يستم تنفيذ هذا الحدث؟ هل سيكون ذلك عند استخدام خصائص
لتفجير الاحداث؟ Visual Basic وطرق الكائن آل مرة؟ ام عندما تزداد شهوة
في داخل RaiseEvent والجواب في أي وقت تريده عن طر يق استخدام العبارة
CPersonالفئة. أآتب هذا الكود في الفئة :

```
Option Explicit
`تعريف حدث جديد
Event DataHasBeenSent)objTo As CPerson, bSuccess As Boolean(
`تعريف طريقة جديدة في الفئة
Public Sub SendData)objTo As CPerson(
```

159

```
`هنا سنقوم بتفعيل الحدث
If objTo Is Nothing Then
RaiseEvent DataHasBeenSent)objTo, False(
Else
RaiseEvent DataHasBeenSent)objTo, True(
End If
End Sub
```

والان ننتقل الى الجهة الاخرى ومعرفة آيفية التفاعل مع هذا الحدث آما تتفاعل
وغيره ا، تتم العملية بنفس الطريقة التي تتعامل Click مع احداث الادوات الاخرى آ
مع الادوات شريطة:
مع تعريف الكائن. WithEvents - استخدام الكلمة المحجوزة
- لا يكون الكائن تابع لمصفوفة.
Public- ان يتم الاعلان عن الكائن على مستوى الوحدة -أي .
وسنح اول الاستجابة لاحداث ذلك CPerson من الفئة Caller سننشئ أسناد باسم
الكائن، في نافذة النموذج أآتب هذا الكود:

```
Option Explicit
Dim WithEvents Caller As CPerson
Private Sub Form_Click()
Dim Khaled As CPerson
Set Khaled  =New CPerson
"خالد" = Khaled.sName
Caller.SendData Khaled
```

End Sub
Private Sub Form_Load()
Set Caller =New MyClass
End Sub

160

Private Sub Caller_DataHasBeenSent)objTo As CPerson, bSuccess As Boolean(
If bSuccess Then
MsgBox"تم ارسال البيانات بنجاح الى: " CPerson.sName
Else
MsgBox "لم اتمكن من ارسال البيانات"
End If
End Sub

ملاحظة: لا يوجد داعي لكتابة اسم الحدث الطويل بلوحة المفاتيح، فبمجرد انتقل الى القائمة WithEvents تعريفك للسطر الذي توجد به آلمة في اعلى يسار نافذة التحرير حتى ترى Combo Box المنسدلة مع باقي اسماء الادوات الموجود ة في MyCallerاسم الكائن النافذة.
، الذي Caller_DataHasBeenSent مثال مبسط جدا يوضح طريقة الاستجابة للحدث SendDataيتم تفجيره بمجرد نجاح الطريقة .

القاء الاحداث:

تستطيع تطبيق مبدأ القاء الاحداث WithEvents عن طريقة الكلمة المحجوزة ، وهي عملية رمي الاحداث من أسناد او اداة الى فئة أسناد Event Multicasting آخر بمجرد تفجير الحدث وقبل ان تنفيذ آلواده . سأوضح الفكرة بالمثال القديم الموجود في الفصل الثاني "النماذج والادوات " وبالتحديد عند فقرة "السيطرة على "، تلاحظ ان الكود المستخدم TextBoxالمدخلات" التابعة لفقرة "أداة النص منذ البداية - انه من غير–للسيطرة على ا لمدخلات أن طويل جدا، وقد اتفقنا المعقول استخدام آل هذه الآواد للتحقق من القيمة التي يكتبها المستخدم في خانة النص، ولكن هنا سنستخدم الكود مرة واحدة فقط، وسنضعه في فئة باسم :CNumTextBox

'الاداة التي ستلقي احداثها الينا
Public WithEvents TextControl As TextBox
'آواد تمنع المستخدم من كتابة الا الارقام
Private OldText As String
Private OldSelStart As Long

161

Private Sub TextControl_GotFocus()
'عندما يكون الترآيز على الاداة
'لابد من حفظ قيمتها
OldText =TextControl.Text
OldSelStart =TextControl.SelStart
End Sub
Private Sub TextControl_Change()
'متغير يمنع استدعاء الاجراء تراجعيا
Static bExitNow As Boolean
If bExitNow Then Exit Sub
If Not IsNumeric)TextControl.Text (Then
'المفتاح المدخل ليس رقم
'قم باعادة عرض القيمة القديمة
bExitNow =True
TextControl.Text =OldText

```
bExitNow =False
TextControl.SelStart =OldSelStart
Else
`القيمة المدخلة رقمية اذا
`قم بحفظها
OldText =TextControl.Text
OldSelStart =TextControl.SelStart
End If
End Sub
Private Sub TextControl_KeyDown)KeyCode As Integer, Shift As Integer(
OldSelStart =TextControl.SelStart
End Sub
Private Sub TextControl_KeyUp)KeyCode As Integer, Shift As Integer(
OldSelStart =TextControl.SelStart
End Sub
Private Sub TextControl_MouseUp)Button As Integer, Shift As Integer, _
```

162

```
X As Single, Y As Single(
OldSelStart =TextControl.SelStart
End Sub
Private Sub TextControl_Click()
OldSelStart =TextControl.SelStart
End Sub
```

جديدة لا تقبل الا الاعداد، فلا يوجد TextBox والان في آل مرة تريد انشاء اداة نص داعي لكتابة آل الآواد السابقة، وانما قم بالقاء جميع احداث اداة النص الى الفئة:

```
Dim NumText As New CNumTextBox
Dim NumText2 As New CNumTextBox
Private Sub Form_Load()
Text1 ="0"
Text2 ="0"
Set NumText.TextControl =Text1
Set NumText2.TextControl =Text2
End Sub
```

لا Text2 و Text1 تمكنا ببساطة شديدة في الكود السابق من جعل الاداتين Event Multicastingتقبلان الا اعداد بفضل القاء الاحداث .

مثال تطبيقي

بامكانك تطبيق مئات الامثلة وانشاء مئات الفئات حتى تجعل حياتك اسهل، الا انني سأآتفي بتطبيق مثال بسيط جدا يتعامل مع الملفات الثنائية.

CFileالفئة :

قد تتعامل آثيرا مع الملفات الثنائية والتي تتطلب دقة في استخدام دوالها، عباراتها واوامرها، وبكل تأكيد الاخطاء الصغيرة تسبب الى تغيير هيئة الملف مما تمكننا من تحرير الملفات CFile ينتج عنه شوائب واخطاء وقت التنفيذ . سنصمم فئة الثنائية بطريقة اسهل، فبدلا من كتابة هذه الآواد المعقدة:

163

```
Dim iFree File As Integer
iFreeFile =FreeFile
Open "MyFile.TXT" For Binary As #iFreeFile
`للكتابة الى الملف
Put #1 , , "اسلوب اجرائي مقرف!"
`للقراءة
Dim sTemp As String
```

```
sTemp =String )18, " "(
Get #1, , sTemp
Print sTemp
```

ما رأيك بكتابة هذه الآواد:

```
Dim MyFile As New CFile
MyFile.OpenFile "MyFile.TXT"
MyFile.PutData "اسلوب أسنادي جميل"
Print MyFile.GetData )16(
```

لا يقتصر الفرق بين الاسلوب الاجرائي الاول والاسلوب الكائني الثاني على اختصار عدد سطور الا آواد فقط، بل حتى في حالات نسيان كتابة الاوامر الضرورية، فمثلا مما يؤدي الى احتجاز مساحة Close تلاحظ انني لم اغلق الملف باستخدام الامر بالذاآرة، اضف الى ذلك احتجاز رقم الملف وعدم امكانية استخدامه لفتح ملف آخر .
اما مع الاسلوب الكائني، فلا يوجد داعي لان اغلق الملف باستدعاء الطريقة تقوم باغلاق ملفاتها تلقائيا بمجرد موت CFile ، لان الكائنات من النوع CloseFile CFileالكائن، فهذا الكود قد اضفته في حدث التدمير للفئة :

```
Private Sub Class_Terminate()
Me.CloseFile
End Sub
```

164

الخاصة باغلاق الملف: CloseFile الذي يقوم باستدعاء الطريقة

```
Public Sub CloseFile()
If Me.iFileNum Then
Close #iFileNum
m_iFileNum =0
End If
End Sub
```

وقد تضيف CFile.CLS ستجد الكثير من الخصائص التي قد اضفتها في الملف التي تحدد lCursorLoc عشرات الخصائص بقدر ما يحلو لك . خذ مثلا هذه الخاصية موقع مؤشر القراءة والكتابة من والى الملف:

```
Public Property Get lCursorLoc ()As Long
lCursorLoc =Seek)Me.iFileNum(
End Property
Public Property Let lCursorLoc)ByVal lNewValue As Long(
Seek Me.iFileNum, lNewValue
End Property
```

هذه امثلة على استخدامها:

```
MyFile.lCursorLoc =1 'بداية الملف
MyFile.lCursorLoc =MyFile.lLOF 'نهاية الملف
```

CFileحتى تحصل على الانجاز الكامل للفئة . Codes.ZIP راجع الملف

165

استخدام الكائنات

تعرفت في الصفحات السابقة على الفئات والفكرة منها وآيفية بنائها، والا ن حان دور استخدام الفئات وانشاء الكائنات منها والتعرف على بعض التفاصيل المتعلقة بالكائنات.

عبارات وآلمات خاصة بالكائنات

من الضروري التعرف على العبارات والكلمات المحجوزة الخاصة بالكائنات حتى تستخدمها الاستخدام الامثل، نبدأ مع انشاء الكائنات باستخدام الكلم ة المحجوزة :New

Newالكلمة المحجوزة :

Visual منه، يمكنك Instance قبل استخدام الكائن عليك بكل تأكيد انشاء نسخة
ولن اتحدث GetObject وCreateObject من انشاء الكائنات بدالتين هما Basic
"، والطريقة الاخرى COM 1 عنهما الا في الفصل الثاني عشر "برمجة المكونات
سواء مع New التي تمكنك من انشاء الكائنات هي باستخدام الكلمة المحجوزة
Setتصريح الكائن او مع العبارة :

```
Dim Turki As New CPerson
Dim Khaled As CPerson
Set Khaled =New CPerson
```

من الاشياء العجيبة جدا جدا والتي تغالط المنطق البرمجي، ان عملية انشاء الكائن
مثل الكود Dim أي مع عبارة–في نفس وقت التصريح New باستخدام الكلمة
التابع للكائن ! ولن يتم Class_Initialize السابق- لا تؤدي الى تفجير حدث الانشاء
بالنسبة لي - هو–تفجيره حتى تستخدم الكائن في آاوادك . والسبب الغريب جدا
لن يقوم فعليا بانشاء الكائن حتى تذآره وتستخدمه في آاوادك، Visual Basic ان
عند التصريح لانشاءه! New بالرغم من اننا استخدمنا الكلمة المحجوزة

Setالعبارة :

في العادة لاسناد أسناد الى آخر: Set تستخدم العبارة

```
Set Khaled =Turki
```

166

عند اسناد الكائنات، لانك ان لم تستخدمها قد Set من الضروري استخدام العبارة
و Turki تظهر لك ر سالة خطأ او حتى نتائج غير متوقعة، فلو افترضنا ان الكائنين
، وقمت باسناد قيمة الكائن الاول sName لهما خاصية افتراضية واحدة هي Khaled
Setالى الثاني دون استخدام العبارة :

```
Khaled =Turki
```

فانك في الحقيقة لم تسند الا قيمة خاصية الكائن الاول الافتراضية الى الخاصية
الافتراضية للكائن الثاني، أي ان حقيقة الكود السابق هي:

```
Khaled.sName =Turki.sName
```

Isالمعامل :

تستخدم هذا المعامل لمعرفة ما اذا أنا المتغيران يشيران الى نفس الكائن:

```
If Khaled Is Turki Then ...
```

ويمكنك ايضا معرفة ما اذا أن الكائن حي يرزق ويتبع لفئة او لا:

```
If Khaled Is Nothing Then ...
```

اما معامل المساواة فارجو ان تنسى فكرة استخدامه للتحقق من مساواة
الكائنات، فالكود التالي:

```
If Khaled =Turki Then ...
```

لا يقارن الا الخصائص الافتراضية -ان وجدت- للكائنات، وأنك أكتبت:

```
If Khaled.sName =Turki.sName Then ...
```

TypeOf ... Isالعبارة :

تمكنك هذه العبارة من اختبار نوع الفئة التي تمثل الكائن:

```
If TypeOf Turki Is CPerson Then ...
```

167

```
If TypeOf MyCtrl Is TextBox Then
MyCtrl.Text ="..."
Else
MyCtrl.Caption ="..."
End If
```

TypeNameالدالة :

تعود بقيمة حرفية تمثل اسم الفئة التابع لها الكائن: TypeName الدالة

```
Print TypeName)Turki (' CPerson
Print TypeName)Text1 (' TextBox
```

القيمةNothing :

اليه: Nothing بامكانك الغاء الكائن في أي وقت بمجرد اسناد القيمة
Set Khaled =Nothing
المزيد من التفاصيل حول موت الكائنات ستقرأها قريبا.

ماهي حقيقة الكائن؟

في البداية اود ان اعرف ماهو الكائن -متغير الكائن ان صح التعبير - يا ترى؟ والجواب
بكل بساطة : متغير الكائن عبارة عن منطقة موجودة في الذاآرة تحمل بيانات
تتعلق بذلك الكائن . قد تكون اجابة السؤال مستنبطة من مبدأ تعريف التراآيبات
% ! ف متغير الكائن مهما 100 الا ان الاجابة السابقة مع الاسف الشديد خاطئة UDT
بت) لانه عبارة عن مؤشر 32 بايت (في نظم 4 أن نوعه فان حجمه لا يزيد عن
الى منطقة في الذاآرة تحمل بيانات تتعلق بذلك الكائن والدليل راقب هذا الكود:
Dim X As New MyClass
Dim Y As New MyClass
'المؤشران
' X, Y
'يشيران الى نفس الكائن
Set Y =X

168

Y.Value =100
' Y.Value =100
Print Y.Value
X.Value =200
' Y.Value =200 !!
Print Y.Value

فان المنطقة من الذاآرة التي أن يشير لها Set Y =X عندما يتم تنفيذ السطر
Yاصبحت نفس المنطقة التي يشير لها المتغير -الكائن- Xالمتغير -المؤشر-
فان X.Value =200 والدليل على ذلك، انني عندما قمت بتعديل قيمة الخاصية
متغيران (أسنادان، مؤشران) Y و X تأثرت بسبب التعديل وذلك لان Y.Value الخاصية
X يشيران الى نفس المنطقة من الذاآرة التي تحتوي على بيانات تتعلق بالكائن
وليس منطقتين مختلفتين. وبكل تأكيد ستسأل نفسك عن المنطقة التي أتت
ما هي اخبارها يا ترى؟ اخبارها يا قارئي العزيز في المشمش ! Y مخصصة للكائن
لانها قد اختفت من الذاآرة وانتهت أي بعبارة لغوية ماتت وستعرف السبب لاحقا.

صورة الكائن في الذاآرة

الكائنات ليست آالمتغيرات العادية فهي تحجز لنفسها منطقتين في الذاآرة الاولى
بايت - والثانية خاصة لبيانات 4 خاصة لمؤشر بيانات الكائن في الذاآرة -حجمه
فلا نحتاج الا Long او Integer الكائن نفسه . اما عندما تعلن عن متغير عادي آ
String لمنطقة واحدة بالذاآرة خاصة بقيمة المتغير باستثناء المتغيرات من نوع
فهي تحتاج الى منطقتين من الذاآرة مثل الكائنات.
Instance Data المنطقة الثانية التي يحتجزها الكائن تعرف بمنطقة نسخة البيانات
، وقسم خاص VTable والمقسمة الى ثلاثة اقسام : قسم خاص بالمؤشر Area
. وفيما يلي Data Area ، وقسم خاص بالمتغيرات التابعة للكائن Counter بالعداد
تفاصيل هذه الاقسام:

المؤشرVTable :

و COM بقدر ما تهم مبرمجي Visual Basic تفاصيل هذا المؤشر لاتهم مبرمجي
عبارة عن مؤشر الى تراآيب في منطقة VTable ++ ، لكن ما استطيع قوله هو ان Co
- وبداية آل اج راء حتى Methods اخرى بالذاآرة تمثل مواقع تنفيذ الاجراءات -الطرق

169

التابعة للكائنات تشير الى نفس التراآيب VTable يتم تنفيذ آاواده . آل مؤشرات

مستقل خاص VTable في حالة آون الكائنات من نفس الفئة . فلكل فئة جدول VTable باجراءات تلك الفئة، ولكل أسناد مؤشر خاص به يشير الى الترآيب المتوافق مع الفئة المنشأ منها، فهنا:

Dim X As MyClass, Y As MyClass, Z As YourClass

VTable تشير الى ترآيبين Z و Y و X خاصة للكائنات VTable توجد ثلاث مؤشرات YourClassو . MyClass خاصين للفئتين

Counterالعداد :

بايت وهو عبارة عن عداد يمثل عدد 4 القسم الثاني من هذه المنطقة حجمه المؤشرات التي تشير الى هذه المنطقة . يبدأ العداد بالقيمة واحد عندما تنشئ الكائن، ويزيد آلما وجد مؤشر اخر يشير الى ذلك العداد . عندما يصل العداد الى الصفر (أي لا يوجد مؤشر يشير الى تلك المنطقة) فان المنطقة يتم تحريرها من الذاآرة وتختفي، وهذا جواب واضح للسؤال متى يموت الكائن؟ راقب هنا:

```
Dim X As MyClass, Y As MyClass
'العداد يبدأ بواحد
Set X =New MyClass
'والان العداد باثنان بسبب
'وجود مؤشران يشيران الى
'نفس المنطقة
Set Y =X
'العداد الان ينقص بواحد
Set X =Nothing
'العداد الان بصفر مما يؤدي الى
'موت الكائن
Set Y =Nothing
```

Data Areaمنطقة البيانات :

والستاتيكية Public وهي المنطقة التي تحتوي على جميع المتغيرات العامة الخاصة بالكائن، بكل تأكيد يختلف حجمها من أسناد لآخر بالاعتماد على عدد Static

170

Longوحجم المتغيرات التابعة له . فمثلا، اذا احتوى الكائن على متغيرين من نوع ، بايت. 8 فان حجم هذا القسم هو

مثال توضيحي:

اود ان اوضح الاقسام الثلا ثة بمثال مع شكل توضيحي له . بافتراض ان لدينا فئة بالاضافة الى MyMethod2 و MyMethod1 تحتوي على طريقتين MyClass باسم . فالكود التالي: Value2 و Value1 باسم Public متغيرين عامين

```
Dim X As MyClass, Y As MyClass, Z As MyClass
'حجز وانشاء نسخ للكائنات في الذاآرة
Set X =New MyClass
Set Y =X
Set Z =New MyClass
'تذآر ان آلاهما مؤشران لمنطقة واحدة
' X, Y
"ترآي" = X.Value1
"العامري" = X.Value2
'اخيرا تعيين قيم للكائن
' Z
"عباس" = Z.Value1
"السريع" = Z.Value2
```

:1- 5يمكن ان نوضح المنطقة الخاصة بالكائنات آما في **الشكل**

171

بالذاآرة. Z و X ،Y : رسم توضيحي لصورة الكائنات**1- 5شكل**

، ولكن استيعابها مسألة1- 5 قد لا تهمك آثيرا المربعات الموجودة في الشكل
عنوان الفقرة التالية - او حت ى الاحتراف في– Binding ضرورية لتعريف فكرة الربط
وهو ما سنتطرق إليه في الفصول اللاحقة. COM برمجة مكونات

Bindingالربط

الكائن سواء أتت خصائص او طرق Members عملية الربط هي باختصار ربط اعضاء
. من VTable بالمؤشر الذي يمثل الكائن وتحديد مواقع الاجراءات في الجدول
لكن عملية VTable المعروف ان الطرق موجودة في مواقع في الذاآرة في الجدول
الوصول لها ليست مباشرة احيانا، راقب هنا:

```
Dim X As Object
If Y =True Then
Set X =New MyClass
Else
Set X = New YourClass
End
X.MyMethod
```

172

، ولكن ماه وX التابعة للكائن MyMethod في السطر الاخير قمت باستدعاء الطريقة
؟ لا نستطيع معرفة ذلك YourClass ام الفئة MyClass ؟ هل هو تابع للفئةX الكائن
، لانه يحتاج الى معرفة Visual Basic الا في وقت التنفيذ حتى تتضح الامور ل
او لا،والتي MyMethod ما اذا أن يوجد به دعم للطريقة VTable ترآيب الجدول
بدورها ستأخذ وقت اطول بكثير من الو صول الى اجراء لكائن معرف النوع سابق ا.
لذلك، اتكلم عن نوعين من انواع الربط هما:

Early Bindingالربط المبكر :

والتي تقوم بتحديد Compiling time عملية الربط المبكر تتم في وقت الترجمة
مما يؤدي الى وصول اسرع بكثير لاعضاء الكائن . طبعاVTable مواصفات الترآيب
لعمل ذلك، لابد من ان تحدد بوضوح نوع الفئة التي سيمثلها الكائن.

```
`التصريح الواضح لنوع الكائنات
Dim X As MyClass, Y As YourClass
Set X =New MyClass
`بافتراض وجود اتصال
` COM
`طبعا ابطأ لكن يعتبر ربط مبكر ايضا
Set Y =CreateObject )"YourServer.YourClass"(
```

Late Bindingالربط المتأخر :

في وقت التنفيذ في آل مرة تصرح فيها VTable هنا يتم تحديد مواصفات الترآيب
المناسب للكائن، VTableعن أسناد جديد، مما يؤدي الى بطء في تحديد الجدول
او Object والتحقق من وجود الطرق المستدع اه. المتغيرات المعرفة من نوع
هي متغيرات لن تستطيع ان تربطها الا عن طريق الربط المتأخر. Variant

```
`تصريح غير واضح للكائنات
Dim X As Object, Y As Variant
Set X =New MyClass
`بافتراض وجود اتصال
` COM
Set Y =CreateObject )"MyServer.YourClass"(
```

173

ولادة وموت الكائن

New ولادة الكائن هي اللحظة التي تنشئ الكائن به ا باستخدام الكلمة المحجوزة
، ويموت الكائن GetObject و CreateObject او الدوال الاخرى التي لم اتطرق لها
بمجرد تحرير المنطقة الخاصة به في الذاآرة آخروجه عن مجاله او تعيين القيمة

له. اعرض لك بعض التفاصيل الدقيقة والخاصة عن انشاء وانهاء الكائن: Nothing
انشاء الكائن واستخدامه:
بحجز منطقة في الذاآرة Visual Basic عندما تقوم بانشاء الكائن لاول مرة يقوم
Visual Basic للفئة التي تمثل ذلك الكائن . بعد ذلك، يقوم VTable تمثل التراآيب
Instancing Data Area بحجز منطقة اخرى بالذاآرة خاصة بمنطقة نسخة البيانات
والتي يقوم بتقسي مها الى ثلاثة اقسام ومن ثم تعبئة المعلومات المطلوبة في
VTable للمؤشر VTable مكانها المناسب في آل قسم . القسم الاول لوضع عنوان
Class_Initialize والثاني يبدأ عداده . ولا يبدأ في القسم الاخير الا بعد تفجير الحدث
VTableلان ذلك الاجراء اصبح عنوانه معروف بفضل تعريف المؤشر .
اما في حالة انشاء أسناد مرة اخرى، فان العملية تتم بشكل اسرع وذلك بسبب ان
فهي موجودة VTableلا يقوم بحجز المنطقة بالذاآرة والخاصة ب Visual Basic
وجاهزة لاي أسناد جديد سينش أ من نفس نوع الفئة السابق . اخيرا، تستطيع
استخدام الكائن واستدعاء طرقه و تعيين خصائصه وحتى انتظار احداثه.
نهاية وجود الكائن بالذاآرة:
Class_Terminate بتفجير الحدث Visual Basic عندما يصل العداد الى صفر سيقوم
والخاص بالكائن متيحا لك فرصة اخيرة لعمل أي شئ قبل موت الكائن ومن ثم
من الذاآرة وفقط، أي لا Instancing Data Area يقوم بتحرير منطقة نسخة البيانات
لانها ستكون بالذاآرة VTable بتحرير منطقة التراآيب Visual Basic تتوقع ان يقوم
واعتقد ان هذا سبب واضح في آون عملية انشاء End حتى نهاية البرنامج بعبارة
VTable الكائن مرة اخرى اسرع بكثير من المرة الاولى بسبب عدم ضرورة انشاء ال
من جديد.
. ففي هذا الحدث تستطيع فعل ما Class_Terminate لدي نقطة اخرى حول الحدث
تريد قبل موت الكائن لكن من المهم معرفة انك لن تستطيع اعادة حياة الكائن عن
طريق هذا الحدث . الفكرة ببساطة هي آالانسان عندما يحتضر، فان من رحمه الله
ييسر له الشهادة وقت الاحتضار وينطق بها ومن ثم يموت لكنه لن يستطيع العودة
الى الحياة من جديد -الا بمعجزة الخالق الذي يحيي ويميت بكل تأكيد - اما مع
فرصة اخيرة لعمل ما تريد قبل ان Class_Terminate فيوفر لك الحدث Visual Basic
يموت الكائن لكنك لن تستطيع اعطائه الحياة من جديد.
174
نقطة اخرى حول موت الكائنات -نسأل الله طولة العمر - هي ان الكائنات لها نظام
يمنع موت الكائن اذا ما أن احد اجراءاته قيد التنفيذ . وبمعنى آخر، لنفترض ان احد
مؤشر عام) X (آايد المتغير Set X =Nothing اجراءات الكائن يقوم بقتل نفسه
راقي جدا جدا ويعلم آيف يتعامل مع هذه النوع من Visual Basic فلابد ان تعلم ان
الكائنات!، فسيقوم باسلوب مهذب جدا اعطاء فرصة للكائن حتى ينهي اجراءه الذي
وقتل الكائن . احسن الله Class_Terminate يتم تنفيذه ومن ثم يقوم بتفجير الحدث
عزاآم.

ارسال الكائن بالمرجع او بالقيمة

" وبالتحديد في قسم الاجراءات BASIC تحدثت في الفصل الثالث "لغة البرمجة
والدوال عن الفرق بين ارسال مرجع المتغير الى الاجراء وارسال قيمة المتغير الى
الاجراء، وذآرت بان المتغيرات المرسلة بالمرجع يمكن لك التعديل في قيمها من
نفس الاجراء، ولكن عند الحديث عن الكائنات فحاول نسيان الفرق بين الارسال
الكائن Pointer بالمرجع والقيمة، لان الكائن في آلا الحالين سيرسل مؤشر
وسيتمكن الاجراء من تعديل جميع محتويات الكائن.
فهو ByRef والكلمة المحجوزة ByVal اما الفرق بين استخدام الكلمة المحجوزة
يؤدي الى انشاء ByVal فرق تقني بحت، اذ ان ارسال الكائن بالكلمة المحجوزة
التابع لمنطقة بيانات Counter نسخة جديدة من المؤشر تؤدي الى زيادة العداد
فان المؤشر هو نفس المؤشر الذي ByRef الكائن، اما الارسال بالكلمة المحجوزة
ارسل الى الاجراء، هذا الكود قد يوضح الفرق:

```
'الارسال بالمرجع ( Sub MySub )objPerson As Person
'تؤدي الى موت الكائن المرسل Set objPerson =Nothing
End Sub
'الارسال بالقيمة ( Sub MySub )ByVal objPerson As Person
'لا تؤدي الى موت الكائن المرسل Set objPerson =Nothing
End Sub
```

175

الفصل السادس

تعدد الواجهات والوراثة

تحدثت في الفصل السابق عن الفئات والكائنات، وذآرت انه آلما أتت الفئة مستقلة آلما زادت امكانية اعادة استخدامها في تطبيقات اخرى، الا انك في حالات آثيرة تود توزيع الآواد بين عدة فئات وتحاول تطبيق روابط بين الفئات لتوفر عليك عناء اعادة كتابة الآواد المتكررة وتسهيل حياتك البرمجية بشكل افضل . في هذا الفصل سأتطرق الى مواضي ع متقدمة في البرمجة أسنادية التوجه واتحدث عن مبدأ تعدد الواجهات ومبدأ الوراثة، واختم الفصل بالتحدث عن فكرة الاهرام الكائنية.

تعدد الواجهات

من المبادئ التي لابد من توفرها في أي لغة Polymorphism مبدأ تعدد الواجهات يدعم الفكرة الاساسية Visual Basic . ومن حسن الحظ OOP برمجة أسنادية التوجه من هذا المبد أ، سأشرح في هذه الفقرة طريقة تطبيق مبدأ تعدد الوجهات آما Abstract Classesاتطرق الى الفئات المجردة .
التعريف البرمجي لمبدأ تعدد الواجهات هو : اسماء متشابهة لكن انجازات مختلفة . المقصد من ذلك، اننا نستطيع Same names but different implementations استدعاء طرق وخصائص متشابهة الاسم لفئات مختلفة البنيان أي بانجازات ، آلا الفئتين CCar والثانية CPerson مختلفة. مثال، نفترض ان لدينا فئتين الاولى ، وبالتالي نستطيع استدعاء Move يوجد بهما طريقة خاصة بالتحريك تسمى . لكن القضية هنا ان عملية CCar.Move و CPerson.Moveالطريقتين باسم ائهما: انجاز الطريقة مختلفة رغم تشابه اسمائها، فمن المعروف ان الشخص يتحرك عن طريق قدميه اما السيارة فبلا شك تتحرك عن طريق الاربع عجلات بها، وهذا هو مبدأ تعدد الوجهات.
المزايا التي تجدها من تعدد الواجهات آثيرة ولعل الميزة الحقيقية هي انها تختصر وغيره ا. فقد تلاحظ ان مبدأ Select Case عليك الكثير من مئات جمل الشرط آ تعدد الواجهات مطبق في الادوات التي تضعها على نافذة النموذج وذلك بسبب وغيره ا، Name او Left وجود الكثير من الخصائص المشترآة بين الادوات آخاصية

176

في وسط TextBox فلو طلبت منك احد الايام كتابة اجراء يقوم بمحاذاة اداة النص النافذة، فستكون حصيلة اصابعك الناعمة الكود التالي:

```
CenterTextBox ) txtTextBox As TextBox (
txtTextBox.Move ScaleWidth -txtTextBox.Width / (2, _
ScaleHeight -txtTextBox.Height / (2
End Sub
```

ولو أتت علاقتنا حميمة جدا وطلبت منك اجراء آخر يقوم بمحاذاة اداة العنوان ، فاعتقد انك ستكتب الاجراء التالي: Label

```
CenterLabel )lblLabel As Label(
lblLabel.Move ScaleWidth -lblLabel.Width / (2, _
ScaleHeight -lblLabel.Height / (2
End Sub
```

اجراء اخر لتوسيط الادوات الثمانية عشر 18 ولا اعتقد انك على استعداد لكتابة

الاخرى حتى لو أتت علاقتنا عاطفية ! بل ستكون مبرمج أسنادي التوجه آآثر وتكتب اجراء واحد يمكن ان يستقبل أي اداة مهما أن نوعها:

```
CenterControl  )ctrlControl As Control (
ctrlControl.Move ScaleWidth  -ctrlControl.Width / (2, _
ScaleHeight  -ctrlControl.Height / (2
End Sub
```

من الاجراء السابق يتضح لنا جمال، قوة، ابداع، مرونة، فن، وسحر مبدأ تعدد الواجهات فالفئة TextBox لها واجهة اخرى باسم Control تحتوي على الطريقة Move الواجهة . هي عبارة عن فئة لكنها Control حالها آحال جميع الادوات الاخرى Abstract Class Control لا تحتوي على اية آآواد، لذلك تسمى فئة المجردة ، فحتى تستطيع ان تحقق مبدأ تعدد الوجهات لابد Interfaceوتحتوي على واجهة من وجود فئة مجردة والتي تعرف الواجهة للفئات الاخرى منها.

تطبيق عملي:

والتي تمثل رحلة ITrip والان لنبدأ بالتط بيق، سننشئ فئة مجردة (واجهة) باسم وتعريف طريقة بها لمعرفة التكاليف:


```
Function GetCost)iDistance As Integer (As Integer
'لا تكتب شيئا هنا فهذه مجرد واجهه
End Function
```

بتمييز الواجهة عن الفئة عن طريق OOP **ملاحظة:** جرى العرف عند مبرمجي قبل اسم الفئة، اما الفئات فما زال حرف I اضافة حرف البادئة هو الآآثر شعبية. C البادئة

CCarوالان انشاء فئة اخرى وهي تمثل رحلة بالسيارة لا تنسى ان تسميها ب :

```
'لا بد ان تضيف هذه العبارة حتى
'نستخدم الواجهة التابعة لفئة
' ITrip
Implements ITrip
Private Function ITrip_GetCost)iDistance As Integer (As Integer
'هذه الدالة مأخوذة من واجهة
' ITrip
ITrip_GetCost  =iDistance  *15
End Function
```

ملاحظة: لا يوجد داعي من كتابة الاجراء السابق بنفسك، فبمجرد كتابة الكلمة المحجوزة Implements بامكانك الوصول الى آاف ة اجراءات الواجهة عن طريق الاداة ComboBox الموجودة في اعلى نافذة محرر الآآواد.

CPlaneايضا فئة اخرى تمثل رحلة بالطائرة :

```
'لا بد ان تضيف هذه العبارة حتى
'نستخدم الواجهة التابعة لفئة
' ITrip
Implements ITrip
Private Function ITrip_GetCost)iDistance As Integer (As Integer
'هذه الدالة مأخوذة من واجهة
```



```
' ITrip
ITrip_GetCost  =iDistance  *100 'لاحظ زيادة سعر التكلفة هنا
End Function
```

، ولاستخدامها انتقل الى CPlane و CCar والفئات ITrip انتهينا من تصميم الواجهة نافذة النموذج ثم ضع اداة زر اوامر واآتب هذا الكود:

```
Private Sub Command1_Click()
```

Dim NewTrip As ITrip
الرحلة الان بالسيارة` Set NewTrip =New CCar
Print NewTrip.GetCost)50(
اصبحت بالطائرة` Set NewTrip =New CPlane
Print NewTrip.GetCost)50(
End Sub
ستلاحظ اختلاف التكاليف بين رحلة با لسيارة واخرى بالطائرة ولو أتت المسافة
آيلو متر). 50متشابهه (

الوراثة

- على اشتقاق Derived الفئة المشتقة–هي قدرة الفئة Inheritance الوراثة
- بحيث تتمكن الفئة المشتقة من الوصول Base Class اعضاء فئة اخرى -الفئة الام
الى جميع اعضاء (طرق/خصائص) الفئة الام ، مما يؤدي الى تطوير الفئة الام وآمال
نواقصها. فمثلا لو أن لدينا الفئة س ونريد اضافة الخاصية ص فيها، فلا يوجد داعي
من اعادة بناء الفئة س من جديد، وانما ننشئ فئة ع تحتوي على الخاصية ص
وتكون الفئة ع مشتقة من الفئة س بحيث تمتلك آافة خصائصها الاخرى . لتوضيح
الفكرة، افترض ان لدينا هذه الفئات الثلاث:
Move، طرقها: Age و Name، خصائصها: CPerson) اسم الفئة: 1
ChangeCollege، طرقها: Major، خصائصها: CStudent) اسم الفئة: 2
ChangeDepartment، طرقها: Salary، خصائصها: CWorkman) اسم الفئة: 3
179
مشتقة CStudent و CWorkman سنطبق عليها مبدأ الوراثة الان بجعل الفئتين
قابلتان CWorkman و CStudent . أي ان الفئتان CPerson ووارثة لاعضاء الفئة الام
، لذلك جميع هذه الاآواد صحيحة من CPerson للوصول الى اعضاء الفئة الام
الناحية المنطقية:
"محمد" = CStudent.Name
CStudent.Age =25
CStudent.Move()
"علوم الحاسب" = CStudent.Major
CStudent.ChangeCollege()
"عبدالله" = CStudent.Name
CWorkman.Age =30
CWorkman.Move()
CWorkman.Salary =10,000
CWorkman.ChangeDepartment()
قابلة للوصول الى CStudent و CWorkman والسبب في ذلك، ان الفئات المشتقة
لا CPerson ، والعكس غير صحيح! فالفئة الام CPerson جميع عناصر الفئة الام
تستطيع الوصول الى اعضاء الفئة المشتقة منها، فلا تكتب في احد الايام شيئا من
هذا القبيل:
CPerson.Salary =20,000
`او
CPerson.ChangeCollege()
المزيد ايضا، الفئات المشتقة ترث من الفئات المشتق منها (الفئات الام) فقط .
لا يمكن لاي فئة منها CWorkman و CStudent ففي مثالنا السابق، الفئات
الوصول الى اعضاء الفئة الاخرى فلا تكتب مثل هذا:
CStudent.Salary =10,000
CWorkman.ChangeCollege()
فقط . وهذا باختصار مفهوم مبدأ الورا ثة في CPerson لانهما مشتقان من الفئة الام
جميع لغات البرمجة والذي يقدم لك الكثير من اختصار كتابة الاآواد والتسهيل في
180

واردت MyClass عملية التنقيح و تطوير الفئة نفسه ا ايض ا، تخيل مثلا ان لديك فئة تطويرها باضافة عناصر جديدة لها، آل ذلك يمكن ان يتم عن طريق اشتقاق فئة اخرى جديدة منها واضافة اللازم.

محاآاة الوراثة بVisual Basic

لا يدعم مبدأ الوراثة بشكل ضمني، والذي سنفعله Visual Basic للاسف الشديد هنا عملية محاآاة مبدأ الوراثة على الفئات . الفكرة في محاآاة الوراثة سهلة ، فبما ان الفئات المشتقة سترث نفس آاواد الفئات الام، فلماذا لا نقوم بنسخ جميع محتويات الفئة الام ولصقها في الفئات المشتقة . لتطبيق ذلك، انشئ فئة باسم واأكتب فيها هذا الكود: CPerson

```
Private m_sName As String
Private m_iAge As Integer
Sub Move()
MsgBox"تم تنفيذ اجراء التحريك"
End Sub
Property Get iAge  ()As Integer
iAge  =m_iAge
End Property
Property Let iAge  )iNewValue As Integer (
m_iAge  =iNewValue
End Property
Property Get sName  ()As String
sName  =m_sName
End Property
Property Let sName  )sNewValue As String (
m_sName  =sNewValue
End Property
```

181

ولصقها CPerson ، قم بنسخ جميع محتويات الفئة الام CStudent ولانشاء فئة ال في الفئة المشتقة:

```
'خصائص الفئة الام
Private m_sName As String
Private m_iAge As Integer
Property Get iAge  ()As Integer
iAge  =m_iAge
End Property
Property Let iAge  )iNewValue As Integer (
m_iAge  =iNewValue
End Property
Property Get sName  ()As String
sName  =m_sName
End Property
Property Let sName  )sNewValue As String (
m_sName  =sNewValue
End Property
'خصائص الفئة المشتقة
Private m_sMajor As String
Property Get sMajor  ()As String
sName  =m_sName
End Property
Property Let sMajor  )sNewValue As String (
m_sName  =sNewValue
```

```
End Property
`طرق الفئة الام
Sub Move()
MsgBox"تم تنفيذ اجراء التحريك"
```

182

```
End Sub
`طرق الفئة المشتقة
Sub ChangeCollege()
MsgBox"تم تنفيذ اجراء تحويل الكلية"
End Sub
```

نفس الفكرة طبقها على الفئة CWorkman :

```
`خصائص الفئة الام
Private m_sName As String
Private m_iAge As Integer
Property Get iAge ()As Integer
iAge =m_iAge
End Property
Property Let iAge )iNewValue As Integer (
m_iAge =iNewValue
End Property
Property Get sName ()As String
sName =m_sName
End Property
Property Let sName )sNewValue As String (
m_sName =sNewValue
End Property
`خصائص الفئة المشتقة
Private m_lSalary As Long
Property Get lSalary ()As Long
lSalary =m_lSalary
End Property
Property Let lSalary )lNewValue As Long (
```

183

```
m_lSalary =lNewValue
End Property
`طرق الفئة الام
Sub Move()
MsgBox"تم تنفيذ اجراء التحريك"
End Sub
`طرق الفئة المشتقة
Sub ChangeDepartment()
MsgBox"تم تنفيذ اجراء تغيير القسم"
End Sub
```

قمنا بعملية محاآاة مبدأ الوراثة، فتستطيع كتابة آواد مثل:__________ والان

```
Dim X As New CStudent
Dim Y As New CWorkman
X.sName = "محمد"
X.iAge =25
X.Move()
X.sMajor = "علوم الحاسب"
X.ChangeCollege()
Y.sName = "عبدالله"
```

```
Y.iAge =30
Y.Move()
Y.lSalary =10,000
Y.ChangeDepartment()
```

علاقة "يحتوي على"

المشكلة في فكرة المحاآاة السابقة هي ضرورة وجود الشيفرة المصدرية Source Code للفئة الام حتى تتمكن من اشتقاق الفئات منه ا. اما في حالة آون الفئة في مكون COM ملف تنفيذي -آداخل مثلا- فالعملية معق دة جدا -ان لم تكن مستحيلة . والحل عن طريق تطبيق علاقة تعرف في عالم البرمجة أسنادية التوجه باسم OOP يحتوي على Has a وهي تنص باختصار على ان الفئة يمكن لها ان تحتوي على

184

أسناد من فئة اخرى عن طريق تعريف متغير يمثل أسناد لتلك الفئة . الان قم باعادة تصميم الفئات المشتقة CWorkman و CStudent بهذه الطريقة:

```
'الفئة الام
Public objPerson As New CPerson
'خصائص الفئة المشتقة
Private m_sMajor As String
Property Get sMajor ()As String
sName =m_sName
End Property
Property Let sMajor )sNewValue As String (
m_sName =sNewValue
End Property
'طرق الفئة المشتقة
Sub ChangeCollege()
MsgBox"تم تنفيذ اجراء تحويل الكلية"
End Sub
```

العيب الوحيد في هذه الطريقة هو ان المستخدم لهذه الفئة لن يستطيع محاآاة الوراثة بشكلها الصحيح، فلن يستطيع كتابة العبارة مثلا X.sName للوصول الى اعضاء الفئة الام، وانما سيضطر الى استخدام الكائن المحضون في الفئة المشتقة وكتابة X.objPerson.sName .

التفويض Delegation

يبدو ان الحل الامثل هو بجعل أسناد الفئة الام مخفي ومحاآاة جميع اعضاءه في الفئة المشتقة، ومن ثم ارسالها الى الكائن، وهذه هي الفكرة الاس اسية من مبد أ التفويض، فسيصبح الكود النهائي للفئة المشتقة CStudent :

```
'الفئة الام
Private objPerson As New CPerson
'تفويض خصائص الفئة الام
```

185

```
Property Get iAge ()As Integer
iAge =objPerson.iAge
End Property
Property Let iAge )iNewValue As Integer (
objPerson.iAge =iNewValue
End Property
Property Get sName ()As String
sName =objPerson.sName
End Property
Property Let sName )sNewValue As String (
objPerson.sName =sNewValue
```

End Property
`خصائص الفئة المشتقة
Private m_sMajor As String
Property Get sMajor ()As String
sName =m_sName
End Property
Property Let sMajor)sNewValue As String (
m_sName =sNewValue
End Property
`تفويض طرق الفئة الام
Sub Move()
objPerson.Move
End Sub
`طرق الفئة المشتقة
Sub ChangeCollege()
MsgBox"تم تنفيذ اجراء تحويل الكلية"
End Sub
186
لكن -مع الاسف - اود ان اخبرك Visual Basic والان قمت بمحاآاة مبدأ الوراثة في % في هذا المثال فقط ! لانه في حالة آون للفئة الام 100 ان الذي فعلناه صحيح - لن تستطيع Polymorphism واجهة فرعية من واجهة اخرى -مبدأ تعدد الواجهات الوصول الى اعضاء الواجهة الاخرى للفئة، والحل تجده في الفقرة التالية.

وراثة الواجهات

اريد ان ابدأ هنا تقديم محتويات المثال الذي سيظهر لنا مشكلة وراثة الواجهات و IMyInterface وآيفية تلافيه ا، سيكون لدينا في هذا المثال واجهة واحدة باسم CDerivedClassوفئة مشتقة باسم . CBaseClass فئة ام باسم
، والفئة الام تحتوي MyMethod تحتوي على اجراء باسم IMyInterface الواجهة
. واخيرا، الفئة المشتقة تحتوي على اجراء باسم BaseMethod على اجراء باسم
، ضع في عين الاعتبار على ان الفئة الام تحتوي على واجهة DerivedMethod
أي ان الطرق التابعة للكائن من الفئة الام هي IMyInterface اضافية من الواج هة
BaseMethodو . MyMethod

توضيح المشكلة:

حتى نقوم بحل المشكلة لابد بكل تأكيد من معرفة ماهي المشكلة . الان سنقوم بعملية التفويض -لمحاآاة الوراثة - آما عملنا في الفقرة السابقة وجعل الفئة ، ليصبح الكود النهائي للفئة CBaseClass م شتقة من الفئة الام CDerivedClass هو: CDerivedClass
Private BaseClass As New CBaseClass
`تفويض طرق الفئة الام
Sub BaseMethod()
BaseClass.BaseMethod
End Sub
`طرق الفئة المشتقة
Sub DerivedMethod()
`أكتب ما تريده هنا
End Sub
187
يبدو ان المشكلة اتضحت لك الان وهي ان الفئة المشتقة لا تدعم الطريقة والتي تعتبر احدى واجهات الفئة الام IMyInterface التابعة للواجهة MyMethod . اذا آنت تفكر بتفويض اجراء لعمل ذلك آهذا: CBaseClass
Sub MyMethod()

BaseClass.MyMethod
End Sub
فارجو ان توقف القراءة في الحال ! لانك بحاجة ماسة الى معرفة و استيعاب مبدأ CBaseClass معرف من الفئة BaseClass ، فالكائن Polymorphism تعدد الواجهات ولن تستطيع عمل ذلك. IMyInterface وليس من الواجهة

حل المشكلة:

اتمنى ان تكون المشكلة قد اتضحت لك ، يكمن الحل بالالتزام بعملية تضم ين في الفئة المشتقة، ويصبح الكود النهائي بهذا الشكل: IMyInterface الواجهة

```
Private BaseClass As New CBaseClass
'لابد من تضمين تلك الواجهة
Implements IMyInterface
'تفويض طرق الفئة الام
Sub BaseMethod()
BaseClass.BaseMethod
End Sub
'تفويض الواجهة الاخرى للفئة الام
Private Sub IMyInterface_MyMethod()
Dim TempInf As IMyInterface
Set TempInf =BaseClass
TempInf.MyMethod 'الان يمكنك عمل ذلك
End Sub
Sub MyMethod()
IMyInterface_MyMethod
End Sub
```

188

```
'طرق الفئة المشتقة
Sub DerivedMethod()
'أكتب ما تريده هنا
End Sub
```

Subclassingالتصنيف الفرعي

OOP بلا شك غير مقبولة بشكل آبير لدى مبرمجي Delegation عملية التفويض وبالتحديد مستخدموا مبدأ الوراثة بكثرة ، الا انها تتميز باعطاءك تحكم آبر قبل تنفيذ الاجراء التابع للفئة الام من داخل الفئة المشتقة . فمث لا، الفئة المشتقة تحتوي على هذا الكود:

```
Sub BaseMethod()
BaseClass.BaseMethod
End Sub
```

وتحتوي على متغيرات Function نفترض ان الطريقة السابقة عبارة عن دالة :Parameters

```
Function BaseMethod )X As Long, Y As Long (As Long
BaseMethod =BaseClass.BaseMethod )X, Y(
End Sub
```

الذي آنت اقصده من التحكم الآبر هو انك في الكود السابق تستطيع الغاء عملية استدعاء الطريقة الموجودة في الفئة الام او تعديل قيم المتغيرات المرسلة Arguments ، اوReturned Value حتى تغيير القيمة التي تعود بها الطريقة فتستطيع ان تكتب شيئا مثل:

```
Function BaseMethod )X As Long, Y As Long (As Long
If X =0 Then
BaseMethod =0
ElseIf Y =0 Then
```

```
BaseMethod =BaseClass.BaseMethod )X, 1(
Else
BaseMethod =BaseClass.BaseMethod )X, Y(
```

189

```
End If
End Sub
```

Subclassing في مبدأ الوراثة، تسمى هذه العملية بالتصنيف الفرعي للفئة الام
، وهي من التقنيات المتقدمة التي توفرها لك لغات البرمجة أسنادية the base class
Visual والتي طبقناها بشكل فعال في لغتنا الجميلة Cآلغة ال ++ OOP التوجه
.Basic
و VB المزيد ايضا، يمكنك تطبيق مبدأ التصنيف الفرعي على جم يع اعضاء مكتبات
حروف الاعداد الستعشرية التي تعود بها تكون انجليزية Hex ، فمثلا الدالة VBA
وقد تكون من المتعصبين الى لغتنا الجميلة بحيث تود ان تعود A، B.... ، F دائما
بتعريف Hex الدالة بالحروف العربية، فتستطيع تطبيق مبدأ التصنيف الفرعي للدالة
BASهذه الدالة في ملف برمجة :

```
Function Hex)lNum As Long (As String
Hex =VBA.Hex$)lNum(
"ا"), "Hex =Replace)Hex, "A
"ب"), "Hex =Replace)Hex, "B
"ت"), "Hex =Replace)Hex, "C
"ث"), "Hex =Replace)Hex, "D
"ج"), "Hex =Replace)Hex, "E
"ح"), "Hex =Replace)Hex, "F
End Function
```

يمكنك استدعائها بنفس الطريقة:

```
Dim lCounter As Long
For lCounter =0 To 15
Print Hex)lCounter(
Next
```

الاهرام الكائنية

عندما تصبح مبرمج أسنادي التوجه، فان نظرتك الى عملية بناء وتصميم البرنامج
تنتطلق من محور آ ئناته وليس آاوده، مجموعة الكائنات التي تصممها تسمى

190

الخاصة ببرنامجك. فلو تنظر الى معظم البرامج Object Hierarchies الاهرام الكائنية
.... الخ او Microsoft Word، Microsoft Excel، Microsoft PowerPoint التجارية آ
.... الخ، تلاحظ ان لكل منتج او عنصر ADO، DAO، DirectX حتى تقنيات اخرى آ
من هؤلاء اهرام أسنادية خاصة به مرتبط بعضها ببعض.
بناء اهرامك الكائنية الخاصة بك امر في غاية الاهمية وآل ما يلزمك هو الترآيز في
تصميم الهرم الكائني وليس في آاوده، فالتصميم الجيد هو العامل الرئيس لنجاح
هرمك الكائني، اما آاوده فتأتي في المرحلة التالية . مع ذلك، اساليب التصميم
واعداد المخططات الاولية لانشاء الهرم الكائني خارج نطاق الكتاب، ولكن سأجهزك
هنا بكل ما تحتاجه لبناء الاهرام الكائنية وسأبدا بالعلاقات بين الفئات.

العلاقات بين الفئات

يوجد نوع من العلاقات تسمى "يحتوي على OOP في عالم البرمجة أسنادة التوجه
" وقد ذآرتها في فقرة "الوراثة" في هذا الفصل، الهدف من هذه العلاقة هو Has a
ربط الفئات بعضها ببعض حتى تتمكن من بناء هرم أسنادي . طريقة الربط تتم بسهولة
شديدة، فكل ما عليك القيام به هو تعريف أسناد من فئة اخرى في داخل الفئة
حتى تصل الى اعضاء الفئة الاخرى. يوجد نوعين من علاقة "يحتوي على" هما:

1: الى 1 علاقة

هي علاقة بين فئة س وفئة ص تربطها أسنادات موجودة في الفئة 1 الى 1 علاقة
واردنا ربطها بالفئة CCar س لتصل الى اعضاء الفئة ص، فمثلا لو أن لدينا الفئة
objOwner بحيث تمثل مالك السيارة وسائق السي ارة فقد تضيف خاصتين CPerson
وتكتب شيئا مثل: CCar في الفئة objDriver و

```
الفئة` CCar
Public objOwner As CPerson
Public objDriver As CPerson
Public sCarModel As String
```

.1- 6وبها تكون قد آونت علاقة في الهرم الكائني البسيط جدا **شكل**
: هرم أسنادي بسيط يبين العلاقة بين أسناداته**.1- 6شكل**

MFCمثل مكتبة ،–فئة 200 طبعا اذا أن الهرم الكائني يحتوي على آآثر من
فسيكون التصميم آما في الشكل السابق امر في غاية الاهمية ويصبح ترآيزك
CPerson وCCar على التصميم آثر من انجاز الاآواد . اما اذا اردت استخدم الفئات
فقد تكتب شيئا مثل:

```
Dim Turki As New CPerson
Dim Abbas As New CPerson
Dim BMW As New CCar
"ترآي العامري" = Turki.sName
"عباس السريع" = Abbas.sName
...
...
BMW.sCarModel ="BMW -7"
Set BMW.objOwner =Turki
Set BMW.objDriver =Abbas
```

وانت في داخل objDriver و objOwner بل يمكنك الوصول الى اعضاء الكائن ات
، فقد تضيف طريقة الى الفئة لطباعة اسم المالك والسائق: CCar الفئة

```
تعريف طريقة في الفئة` CCar
Public Sub PrintRelatedPeople )frmForm As Form(
Me.sCarModel & "الموديل:" frmForm.Print
Me.objOwner & "المالك:" frmForm.Print
Me .objDriver & "السائق:" frmForm.Print
End Sub
```

برمجة ممتعة للغاية واقرب الى العالم الحقيقي OOP فعلا، البرمجة أسنادية التوجه
من البرمجة الاجرائية المعقدة.

الى ن: 1 علاقة

قد تكون هناك آآثر من علاقة بين نفس الكائنات لتمثل علاقة آآثر تعقيدا تعرف
الى ن، حيث يحتوي الكائن على مجموعة أسنادات من نفس النوع . فلو 1 بعلاقة
لها مالك واحد CCar عدنا الى المثال السابق، سنلاحظ ان السيارة الواحدة
، نستطيع تطوير أسناد السائق بحيث يمكن objDriver و سائق واحد objOwner
ولد نعمة ! قد تستخدم اس لوب المصفوفات–للسيارة ان يكون لها آآثر من سائق
آما في الكود التالي: objDriver لتعيد تعريف الكائن

```
الفئة` CCar
Private m_objDrivers )5 (As CPerson
Public Property Get objDriver )iDriverNum As Integer (As CPerson
Set objDriver =m_objDrivers )iDriverNum (
End Property
Public Property Set objDriver )iDriverNum As Integer, ByVal objNewValue As
CPerson(
Set m_objDrivers )iDriverNum = (objNewValue
```

End Property
بامكانك الوصول الى هذه الخاصية بكتابة شيئا مثل:
Set BMW.objOwner =Turki
Set BMW.objDriver)0 = (Abbas
Set BMW.objDriver)1 = (Ahmed
Set BMW.objDriver)2 = (Ali
...
...
Array based 1الى ن مبنية على المصفوفات " 1 يعرف النوع السابق بانه "علاقة
الى ن مبنية 1 افضله آثيرا- يعرف "بعلاقة–، ويوجد نوع اخر to many relationship
، حيث تكون الخاص ية Array based 1 to many relationshipعلى المجموعات "
بدلا من مصفوفة: Collection عبارة عن مجموعةobjDriver
CCar 'الفئة
Public objDrivers As New Collection
مباشرة: objDriver بامكانك اضافة الكائنات الى الخاصية
BMW.objDrivers.Add Turki
BMW.objDrivers.Add Ali
For ... Eachباستخدام حلقة : objDriver او حتى الوصول الى آافة عناصر الخاصية
Dim objDriver As CPerson
For Each objDriver In BMW.objDrivers
Print objDriver.sName
Next
المشكلة في العلاقات المبنية على المجموعات تظهر عندما نعلم ان المجموعة
يمكنها ان تحمل أي نوع من القيم، فهذا الكود سيتم تنفيذه objDriver(الخاصية)
بشكل صحيح:
Dim BMW As New CCar
Dim Mercury As New CCar
...
...
BMW.objDrivers.Add Mercuty 'سائق السيارة هي سيارة اخرى!
CPersons Collection Class والحل يتم بانشاء فئة خاصة تسمى فئة المجموعة
لها طرق وخصائص قياسية يتبعها آل المبرمجين وجميع المكت بات والكائنات
آما في COM و لغات البرمجة الاخرى المتوافقة مع Visual Basic المتوفرة في
الفقرة التالية.

Collection Classes فئات المجموعات

فئات المجموعات ما هي الا فئات عادية لكن لها خصائص وطرق قياسية عليك
، وان اصررت Collection Class أكتباعها حتى يطلق على الفئة اللقب "فئة مجموعة"
على عدم أكتباع هذه المواصفات القياسية للفئة، فارجو ان تعود الى رشدك وتلين
COM عنادك قليلا، فجميع الفئات المنجزة بلغات البرمجة المختلفة والداعمة لتقنية
Visual Basicتتبع هذا الاسلوب بما فيهم .
اول قاعدة عليك معرفتها هي ان فئة المجموعة تمث ل مجموعة لكائنات معينة
يمكن ان تكون تابعة لفئة CPerson "، فالفئةsونمييزها عن فئة الكائنات بالحرف "
CPersonsمجموعة باسم .
واأكتب هذا الكود CPerons انشئ فئة جديدة وسمها CPersons اما لبناء المجموعة
Collectionsالمحاآي لطرق وخصائص المجموعات :
CPersons 'فئة المجموعة
Private m_Col As New Collection
Public Sub Add)objNewItem As CPerson, Optional vKey As Variant, _
Optional vBefore As Variant, Optional vAfter As Variant(
m_Col.Add objNewItem, vKey

End Sub
Public Sub Remove)vIndexKey As Variant(
m_Col.Remove vIndexKey
End Sub
Public Property Get Count ()As Long
Count =m_Col.Count
End Property
Public Property Get Item)vIndexKey As Variant (As CPerson
Set Item =m_Col)vIndexKey(

195

End Property

ملاحظة: تستطيع استخدام الاضافة Add-Ins المسماةClass Builder Utility لتسهيل عملية كتابة طرق وخصائص فئة المجموعة تلقائيا ومن ثم تعديلها بنفسك.

الى Collection من objDrivers وتغيير نوع الكائن CCar بامكانك الانتقال الى الفئة :CPersons

Public objDrivers As New CPersons

بنفس الطريقة السابقة وأن شيئا لم CPerson و CCar والان استخدم الفئات يحدث:

Dim BMW As New CCar
Dim Driver1 As New CPerson
Dim Driver2 As New CPerson
...
...
"محمد" = Driver1.sName
"عبدالله" = Driver2.sName
...
...
BMW.objDrivers.Add Driver1
BMW.objDrivers.Add Driver2

ضرورة الالتزام بالمعايير القياسية:

في بداية الفقرة آادت على مسألة الالتزام بالمواصفات والمعايير القياسية لفئات المجموعات، وقد قمنا بعملية محاآاة لمعظم طرق وخصائص المجموعات آما التابعة Item ينبغي، ولكن بقيت نقطتين بودي توضيحها، الاولى تتعلق بالخاصية Default Property ، فيجب ان تكون الخاصية الافتراضية CPersons للمجموعة بحث يمكن للمبرمج تجاهلها: CPersons للمجموعة

196

Print BMW.objDrivers.Item)1.(sName
Print BMW.objDrivers)2.(sName

آما ذآرت في Procedure Attributes بامكانك عمل ذلك عن طريق صندوق الحوار OOPالفصل السابق "البرمجة أسنادية التوجه ."

مع For ... Each اما النقطة الثانية التي اود ان اذآرها هي قابلية استخدام الحلقة ، فلو أكتبت هذا الكود: CPersons المجموعة

Dim objDriver As CPerson
For Each objDriver In BMW.objDrivers
Print objDriver.sName
Next

ليست مدعومة في For ... Each رسالة خطأ، لان الحلقة Visual Basic سيظهر لك ، لذلك عليك كتابة الكود التالي في المجموعة CPersons مجموعتنا الجديد :CPersons

```
Public Property Get NewEnum ()As IUnknown
Set NewEnum  =m_Col]._NewEnum[
End Property
```

في صندوق NewEnum التابعة للاجراء Procedure ID - في الخانة4 وكتابة القيمة
، Hide this member وتحديد الاختيار Procedure Attributes الحوار
CPersonsمع المجموعة . Each ... For وتكون بذلك قادر على استخدام الحلقة
NewEnum: تعديل مواصفات الاجراء .
ملاحظة: استيعاب الخطوات السابقة خارج نطاق الكتاب، لانه يتطلب فهم
والخاصة OLE Automation التابعة ل COM البنية الترآيبات لمكونات
او بالتحديد الواجهة IUnknown بواجهات المكونات آالواجهة
. اذا اردت مزيد من التفاصيل المتقدمة حول هذا IEnumVariant
الموضوع انصحك بكتاب:
Advanced Visual Basic 6
Power Techniques for Everyday Program
By :Matthew Curland
ISBN :0-201-70712-8
فيفضل CCar و CPerson، CPersons اخيرا، اذا اردت اعادة رسم الهرم الكائني
استخدام لون يميز فئات المجموعات عن الفئات العادي فهو الاسلوب
التي رأيتها.–المتبع في ملفات التعليمات ومواقع الانترنت
CPersons: الهرم الكائني بعد اضافة المجموعة .
حقيقي Visual Basic بهذا آون قد انتهيت من تشييد بنية اساسية لتكون مبرمج
Visual Basic بعدما تطرقت الى المبادئ والاساسيات التي لابد على آل مبرمج
. وهذه نهاية الجزء الاول Visual Basic من معرفتها واتقانها للابحار في برمجة
"الاساسيات" من هذا الكتاب، والان بامكانك تعليم نفسك ذاتيا اما بالحصول على
كتب متخصصة في مجال معين، او قراءة مقالات متقدمة.

تم بحمد الله و نعمته

أعداد المبرمج:مشتاق طالب رشيد العامري

COM.MUSHTAQ_TALIB58@YAHOO